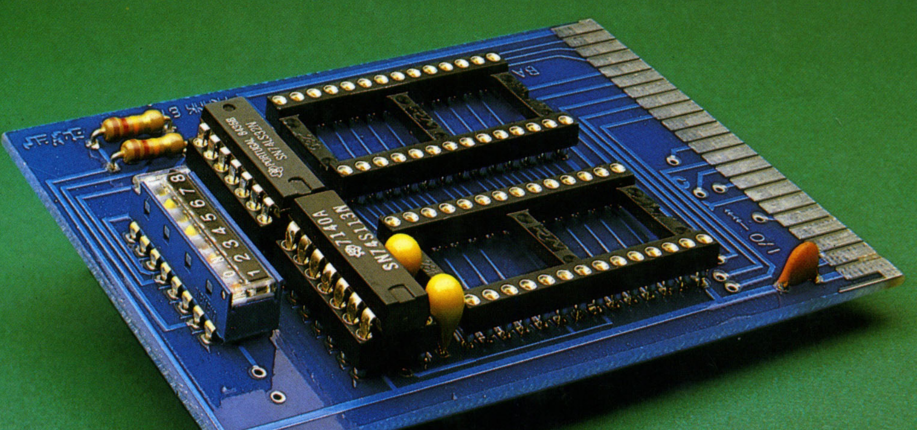


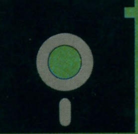
Uwe Gerlach

Hardware-Basteleien zum C 64/C 128



Eine leicht verständliche Einführung in die digitale Schaltungstechnik. Mit vielen Platinenlayouts und ausführlichen Selbstbauanleitungen für einen Sprachausgabebaustein. Radioaktivitätsmeßgerät. 128Kbyte-EPROM-Karte etc.

Mit Treibersoftware zu allen Hardware-Zusätzen.



Uwe Gerlach

Hardware-Basteleien zum C64/C128

- Eine leicht verständliche Einführung in die digitale Schaltungstechnik
- Mit vielen Platinenlayouts und ausführlichen Selbstbauanleitungen für einen Sprachausgabebaustein
- Radioaktivitätsmeßgerät
- 128-Kbyte-EPROM-Karte etc.

Markt&Technik Verlag AG

CIP-Kurztitelaufnahme der Deutschen Bibliothek

Gerlach, Uwe:
Hardware-Basteleien zum C64/C128 : e. leicht verständl. Einf. in d. digitale Schaltungstechnik ; mit vielen Platinenlayouts u. ausführl. Selbstbauanleitungen für e. Sprachausgabebaustein ; Radioaktivitätsmessgerät ; 128-Kbyte-EPROM-Karte etc. / Uwe Gerlach. – Haar bei München : Markt-und-Technik-Verlag, 1987. – & 1 Diskette
ISBN 3-89090-389-4

Die Platinen wurden vom Verlag nachgebaut und mit der veröffentlichten Software intensiv getestet. Soft- und Hardwarefehler konnten nicht festgestellt werden. Trotzdem haftet der Markt&Technik Verlag nicht für Schäden jeglicher Art, die auf den Einbau zurückzuführen sind. Außerdem bedenken Sie, daß durch das Öffnen des Gerätes die Garantie erlischt.
Vorsicht, der Computer wird mit 220 Volt betrieben.
Bei geöffnetem Gerät stets die Stromzuführung unterbrechen.

15 14 13 12 11 10 9 8
90

ISBN 3-89090-389-4

Einbandgestaltung: Grafikdesign Heinz Rauner
Druck: Jantsch, Günzburg
Printed in Germany

Inhaltsverzeichnis

Vorwort über dieses Buch

Der C 64 ist seit einigen Jahren der am weitesten verbreitete Heimcomputer. Angelockt durch bunte Spiele hat er extrem viele Käufer – zumeist Jugendliche – in seinen Bann gezogen. Einen Großteil der Anwender interessiert es nun, nachdem die erste Spielwut verflogen ist, was außer reinem Programmieren noch alles mit einem Computer angefangen werden kann.

Wie aus Umfragen bei Heimcomputerbenutzern deutlich hervorgeht, ist das Interesse an Zusatzhardware zum Selberbauen sehr stark angewachsen. So dient der C 64 unversehens als Grund, sein Hobby auf die Elektronik auszudehnen.

Aus eigener Erfahrung weiß ich, daß der selbständige Entwurf von Digitalschaltungen eigentlich gar nicht schwer ist. Man benötigt nicht etwa große Formeln aus dem Bereich der diskreten Elektronik. Nein, es reicht für den Anfang, fertige Digital-ICs wie Bauklötze aus einem Spielzeugkasten hintereinanderzusetzen. Oft fehlen lediglich die geeigneten Informationsquellen. Man muß sich jedoch erst einmal an das System und seine verschiedenen Elemente gewöhnen.

Dieses Buch ist speziell für Anwender geschrieben, die aus der Kenntnis ihres Computers heraus auch in die Welt der Elektronik einsteigen wollen. Es erläutert eingehend alle vorhandenen Schnittstellen des C 64 und demonstriert ihren Einsatz anhand vieler interessanter Bauvorschläge. Mein Anliegen ist dabei, die mit eigenen Entwicklungen gewonnenen Erfahrungen anhand von ausführlichen, illustrierten Erläuterungen interessierten Hardware-Bastlern allgemein verständlich weiterzugeben.

Die Beschreibungen zu den vorgestellten Hardwarezusätzen sind daher so gehalten, daß auch interessierte Einsteiger den Entwurf Stück für Stück nachvollziehen können. Mehr noch, sie werden in die Lage versetzt, selbst eigene Ideen zu realisieren. Trotzdem sollen auch alte Elektronik-Hasen nicht zu kurz kommen.

Besonderer Wert wurde auf die Nachbausicherheit der vorgestellten Schaltungen gelegt. Zu allen wichtigen Geräten sind Platinenvorlagen abgedruckt, die eine große Hilfe beim Belichten und Ätzen der Leiterplatten darstellen. Diese können dann selbst von blutigen Laien nach den im Buch enthaltenen Anleitungen und Tips sicher bestückt werden. Auch die wenigen doppelseitigen Leiterplatten sind – im Gegensatz zu vielen anderen veröffentlichten Platinen – ohne große Klimmzüge

im Hobbylabor herstellbar. Die Durchkontaktierungen fallen nicht mit den IC-Anschlußbeinchen zusammen, so daß genügend Platz zum Löten auf der Bestückungsseite verbleibt. Doch auch wer die Schaltungen lieber abändern möchte, kommt nicht zu kurz. Zu allen verwendeten Bauteilen enthält das Buch die genaue Anschlußbelegung. Das werden besonders Leser, die mit Lochrasterplatten arbeiten wollen, zu schätzen wissen.

Man muß das Buch nicht unbedingt linear von Deckel zu Deckel durchlesen. Vielmehr ist jede Bauanleitung prinzipiell eine in sich abgeschlossene Einheit. Viele Querverweise zeigen trotzdem Zusammenhänge zwischen den Zeilen auf und geben auch eiligen Lesern schnell einen umfassenden Überblick. Außerdem erlaubt das ausführliche Stichwortverzeichnis einen schnellen Zugriff auf benötigte Informationen.

Voraussetzungen

Hardwarevoraussetzung zum erfolgreichen Nachbau und Betrieb der vorgestellten Selbstbau-Geräte ist – neben einer gut sortierten Werkzeugkiste für den Elektronik- und Feinmechanikbereich (Opas 500-Watt-Lötkolben für die Dachrinne gehört zurück in den Keller!) – ein Commodore 64. Natürlich ist ebenso ein C 128 im 64er Modus möglich. Zu manchen Schaltungen wird zusätzlich noch das Diskettenlaufwerk VC 1541 bzw. ein Kassettenrecorder benötigt.

Es sind nur wenige Vorkenntnisse zum Verständnis dieses Buches nötig. Für absolute Anfänger auf dem Gebiet der Elektronik gibt Anhang A sogar eine detaillierte Einführung in die Kunst des Lötens. Trotzdem ist etwas Erfahrung im Umgang mit dem Lötkolben empfehlenswert. Allein zu wissen, an welchem Ende dieses Gerät heiß wird, hilft noch nicht beim Einlöten von Bauteilen auf engstem Raum. Als Anfänger sollte man probeweise mit dem Verlöten von Drahtenden beginnen und sich nicht gerade die größte Platine als erstes Objekt vornehmen.

Aus Ihren bisherigen Erfahrungen mit Computern wissen Sie sicherlich, daß selbst die besten Geräte nichts bringen, wenn die Software fehlt. Eine weitere Forderung ist daher die Beherrschung der Programmiersprache BASIC. Dagegen sind Maschinensprachkenntnisse nicht unbedingt nötig. Viele Support-Programme benutzen zwar Assembler-Teile, ich habe jedoch darauf geachtet, diese so zu schreiben, daß sie als Unterprogramme von BASIC her aufrufbar sind. So können allein mit Hilfe einer Funktionsbeschreibung der Maschinenroutine auf einfache Weise Änderungen am BASIC-Programm durchgeführt werden.

Diskette liegt bei!

Um Ihnen Tipparbeit zu sparen, liegt diesem Buch eine Diskette mit der Software bei. Sie enthält auch alle Quelltexte und die zugehörigen Object-Files.

Die Quelltexte wurden für den Assembler PROFI-ASS 64 V2.0 (von Data Becker) erstellt. Sie tragen den Zusatz .ASM am Ende des Filenamens, während entsprechende Object-Files die Endung .OBJ aufweisen. BASIC-Lader, bei denen das Maschinenprogramm lediglich in DATA-Zeilen abgelegt wurde, führen das angehängte Kürzel .BAS. Aus Platzgründen wurde auf den Abdruck der Listings verzichtet.

Das erste Programm im Inhaltsverzeichnis ist ein Universallader, der alle direkt startbereiten BASIC-Programme in einem Menü anzeigt und auf Tastendruck aktiviert. Beachten Sie zu jedem Programm die Hinweise im Text.

Falls Sie selbst kein Diskettenlaufwerk besitzen, wird Ihnen sicherlich ein Bekannter die Files auf Kassette kopieren. Die beiliegende Diskette ist nicht kopiergeschützt. Dadurch sind Änderungen nach eigenen Bedürfnissen an einer Sicherheitskopie (!) ohne Schwierigkeiten möglich. Solche Änderungen in den Betriebsprogrammen sind übrigens sehr empfehlenswert, da nur so die Hardware an Ihre speziellen Erfordernisse angepaßt werden kann, und außerdem auf diese Weise die Funktion der Zusatzschaltungen oft erst richtig klar wird.

Zum Schluß dieses Vorwortes möchte ich nicht versäumen, der Firma Valvo für ihre freundliche Unterstützung zu danken, ebenso den Personen, die mich mit Rat und Tat unterstützt haben.

Allen Leserinnen und Lesern wünsche ich nun eine angenehme Lektüre und viel Erfolg beim Nachbauen bzw. eigenen Entwerfen von Zusatzschaltungen.

Uwe Gerlach Bebra-Weiterode

Kapitel 1
Grundlagen der digitalen Schaltungstechnik

Die Digitaltechnik ist für den Elektronik-Neuling ein geradezu idealer Einstieg. Wie bei einem Baukasten kann man fertige Grundelemente beliebig aneinanderreihen, ohne sich große Gedanken über spezielle Kleinigkeiten machen zu müssen. Die Elemente und Regeln dafür sollen in diesem Kapitel vorgestellt werden.

1.1 Das Wesen digitaler Größen

Es gibt im Leben viele Dinge, die nur zwei Zustände haben können. Zum Beispiel kann ein Lichtschalter ein- oder ausgeschaltet sein, ein Klingelknopf kann gedrückt sein oder nicht, die Spitze eines Kugelschreibers kann herausragen oder eingezogen sein. In jedem Fall existieren nur zwei eindeutig unterscheidbare Zustände. Man spricht von digitalen oder binären Größen. Jeden Zustand kann man als Information auffassen. Solche Informationen zu verarbeiten, ist technisch besonders einfach, da lediglich das Fällen von JA-NEIN-Entscheidungen nötig ist – aufgrund von Bedingungen, die entweder vorhanden sind oder nicht.

Wenn zum Beispiel ein Autofahrer rotes Licht oder ein STOP-Zeichen oder ein Hindernis sieht, wird er anhalten. Symbolisch kann die HALT-Bedingung wie in Bild 1.1 dargestellt werden. Jede der drei Bedingungen kann erfüllt sein oder nicht. Ist eine ODER mehrere der Bedingungen

Bedingungen			Ergebnis
Rote Ampel	STOP-Zeichen	Hindernis	HALT
Nein	Nein	Nein	Nein
Nein	Nein	Ja	Ja
Nein	Ja	Nein	Ja
Nein	Ja	Ja	Ja
Ja	Nein	Nein	Ja
Ja	Nein	Ja	Ja
Ja	Ja	Nein	Ja
Ja	Ja	Ja	Ja

Tabelle 1.1 Wahrheitstabelle für die HALT-Entscheidung

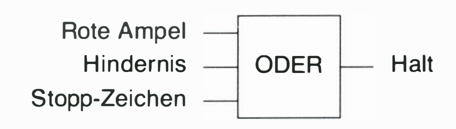


Bild 1.1: Bedingung für das Halten

erfüllt, dann hält der Fahrer an. Tabelle 1.1 zeigt alle möglichen Kombinationen der drei Eingangsbedingungen. Es ist die sogenannte Wahrheitstabelle, s. Seite 17.
Das eben behandelte Beispiel demonstriert eine logische Entscheidung. Sie lautet je nach Kombination der Eingangsbedingungen HALTEN oder NICHT HALTEN. Auch das Ergebnis ist also wieder eine digitale Größe.

1.2 Verknüpfungen

Logische Entscheidungen können von Menschen, aber auch von irgendwelchen mechanischen, hydraulischen oder auch elektronischen Vorrichtungen getroffen werden. Ungeachtet der Arbeitsweise der Vorrichtung, kann die Wirkung symbolisch in einer allgemeingültigen Form dargestellt werden. Im behandelten Beispiel kam eine ODER-Entscheidung vor. Das gängige Symbol dafür zeigt Bild 1.2.

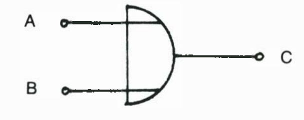


Bild 1.2: Übliches Symbol für die ODER-Verknüpfung

Die Eingänge sind links an das Symbol angetragen und allgemein mit den Buchstaben A und B bezeichnet. Natürlich kann jede Verknüpfung auch mehrere Eingänge haben; im Beispiel des haltenden Autofahrers waren es drei. Der Ausgang ist rechts an das Symbol angetragen. Statt ODER-Verknüpfung sagt man auch englisch OR-Gatter.
Es gibt viele Möglichkeiten, die zwei Zustände einer binären Größe zu bezeichnen. Wir haben bisher die Ausdrücke JA und NEIN benutzt. Daneben sind aber (leider) noch viele andere Bezeichnungen zu finden. In Tabelle 1.2 sind die häufigsten Ausdrücke zusammengestellt:

JA	NEIN
wahr	falsch
true	false
1	0
L	0
HIGH	LOW
H	L

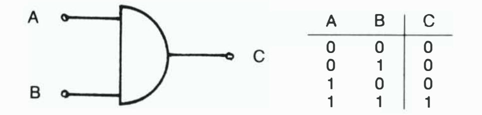
Tabelle 1.2: Die unterschiedliche Nomenklatur

Sie sehen, daß sich manche Abkürzungen sogar überschneiden, was mitunter sehr verwirrend sein kann. Wir wollen deshalb in diesem Buch die Bezeichnungen HIGH oder 1 benutzen, falls die Bedingung erfüllt ist, und LOW oder 0, falls nicht. Diese Namensgebung entspricht den tatsächlichen Verhältnissen bei elektronischen Schaltungen, in denen – wie wir noch sehen werden – mit zwei unterschiedlichen Spannungen gearbeitet wird.
Tabelle 1.3 gibt die Wahrheitstafel der ODER-Verknüpfung aus Bild 1.2 in der üblichen Schreibweise wieder.
Eine andere Art, Entscheidungen zu treffen, erfordert die sogenannte UND-Verknüpfung. Nehmen wir an, unser Autofahrer bekommt grünes Licht UND die Fahrbahn ist frei; erst dann kann er fahren. Es müssen also die eine UND die an-

A	B	C
0	0	0
0	1	1
1	0	1
1	1	1

Tabelle 1.3: Wahrheitstabelle der ODER-Verknüpfung mit zwei Eingängen

dere Eingangsbedingung erfüllt sein. Bild 1.3 zeigt das Symbol der UND-Verknüpfung sowie die zugehörige Wahrheitstabelle. Auch hier sagt man statt UND oft englisch AND.



A	B	C
0	0	0
0	1	0
1	0	0
1	1	1

Bild 1.3: Übliches Symbol für die UND-Verknüpfung und Wahrheitstabelle

Manchmal ist es nötig, eine digitale Information in ihr Gegenteil umzuwandeln. Die Einrichtung, die hierzu verwendet wird, heißt Inverter oder Negierer. Bild 1.4 enthält wieder das Symbol und die ent-



A	B
0	1
1	0

Bild 1.4: Symbol für den Inverter mit Wahrheitstabelle

sprechende Wahrheitstabelle. Es ist üblich, den negierten Wert einer Größe durch einen hochgestellten Querstrich zu kennzeichnen. Der invertierte Wert einer Größe A ist also $\overline{A}$.
Ein Inverter kann zusammen mit UND- oder ODER-Verknüpfungen benutzt wer-

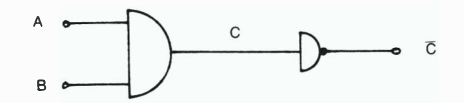
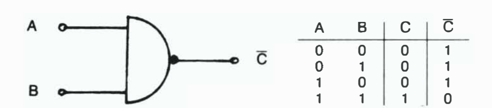


Bild 1.5: UND-Verknüpfung mit nachgeschaltetem Inverter

den, um deren Fähigkeiten zu erweitern. Sehr oft liegt der Ausgang invertiert vor, was aus schaltungstechnischen Gründen einfacher und praktischer ist. Bild 1.5 zeigt, wie sich das Verknüpfungsglied dann zusammensetzt und in Bild 1.6 erkennt man die übliche vereinfachte Darstellung. Der negierte Ausgang wird oft in der englischen Bezeichnung durch ein vor-



A	B	C	$\overline{C}$
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

Bild 1.6: Vereinfachte Darstellung der Schaltung aus Bild 1.5 mit Wahrheitstabelle

angestelltes N gekennzeichnet. Eine UND-Verknüpfung mit negiertem Ausgang wäre also ein NAND, eine entsprechend negierte ODER-Verknüpfung ein NOR.
Natürlich gibt es noch wesentlich mehr Verknüpfungsmöglichkeiten, doch die bisher vorgestellten sind die gängigsten Grundelemente. Aus ihnen lassen sich prinzipiell auch die kompliziertesten Schaltungen aufbauen.

1.3 Integrierte Digitalschaltungen

Die stürmische Entwicklung der Elektronik in den vergangenen drei Jahrzehnten wurde durch zwei bemerkenswerte technologische Durchbrüche regelrecht erzwun-

gen. Da war zu Beginn dieser Zeitspanne die Entwicklung und Verfeinerung der modernen Halbleitertechnologie und bald darauf die Bereitstellung immer konsequenter durchdachter, integrierter elektronischer Schaltungen (englisch: Integrated Circuit, kurz: IC). Gerade diese integrierten Schaltungen haben revolutionär gewirkt, weil sie ganze Funktionskomplexe zusammenfassen. Umfangreiche Anwendungsprobleme können damit preisgünstig, schnell und ohne viel zusätzlichen Aufwand gelöst werden.

1.3.1 Die TTL-ICs

Von den integrierten Schaltungen läßt sich die Untergruppe der integrierten Digitalschaltungen – und hier wiederum die Digitalbausteine der TTL-Familie – besonders leicht auch im Hobbybereich einsetzen. TTL heißt Transistor-Transistor-Logik, und die so gefertigten Bausteine haben die Bezeichnung 74XX, wobei XX für beliebige zwei- oder dreistellige Kennzahlen steht. Die Digitalbausteine der Reihe 74XX sind unkompliziert in der Handhabung, sehr robust (sie nehmen meist sogar eine kurze Verpolung der Betriebsspannung nicht sonderlich übel) und sie sind – das ist besonders wichtig – in allen Elektronikgeschäften zu außerordentlich günstigen Preisen zu haben. Das System der verschiedenen Digitalbausteinfunktionen ist sehr umfangreich. Man kann als Hobbybastler nicht einmal annähernd die Möglichkeiten ausschöpfen, die sich aus den angebotenen Bausteinvarianten ergeben, aber die Baukastenphilosophie bietet eine Chance: Wenn man sich einmal mit den Grundzügen des 74XX-Systems vertraut gemacht hat, kann man der Phantasie bei der Suche nach neuen Schaltungen freien Lauf lassen. Man benötigt keine detaillierten Kenntnisse aus der Elektronik, man muß keine Bauteilwerte dimensionieren und berechnen können – allein auf den logischen Aufbau kommt es an.

1.3.2 Das Zusammenschalten von 74XX-ICs

TTL-ICs der Reihe 74XX benötigen eine stabile Spannungsversorgung von 5 Volt. Zwischen 0 und 5 Volt liegen auch die Ein- und Ausgangsspannungen der Schaltglieder.

Am Eingang eines Digitalbausteins der 74XX-Serie wird ein elektrischer Zustand als LOW-Signal identifiziert, wenn die Eingangsspannung kleiner als +0,8 Volt bleibt. Als HIGH-Signal wird eine Eingangsspannung erkannt, die höher als +2,0 Volt liegt. Der Bereich dazwischen ist undefiniert, da man nicht genau sagen kann, ob das Gatter HIGH oder LOW erkennen wird. Man sollte eine solche Eingangsspannung vermeiden.

Da digitale Bausteine auf digitale Eingangssignale mit digitalen Ausgangssignalen antworten müssen und diese wieder von digitalen Bausteinen weiterverarbeitet werden sollen, garantieren die Hersteller der Reihe 74XX, daß ein LOW-Ausgangssignal kleiner oder gleich +0,4 Volt und ein HIGH-Ausgangssignal größer als +2,4 Volt ist. So wird eine sichere Funktion auch unter massiven Störeinflüssen gewährleistet. Bild 1.7 macht die Verhältnisse noch einmal grafisch deutlich.

Eine Spannung bezeichnet man in der Elektronik auch als Potential. Daher wird in Zukunft oft von HIGH- und LOW-Potential oder von HIGH- und LOW-

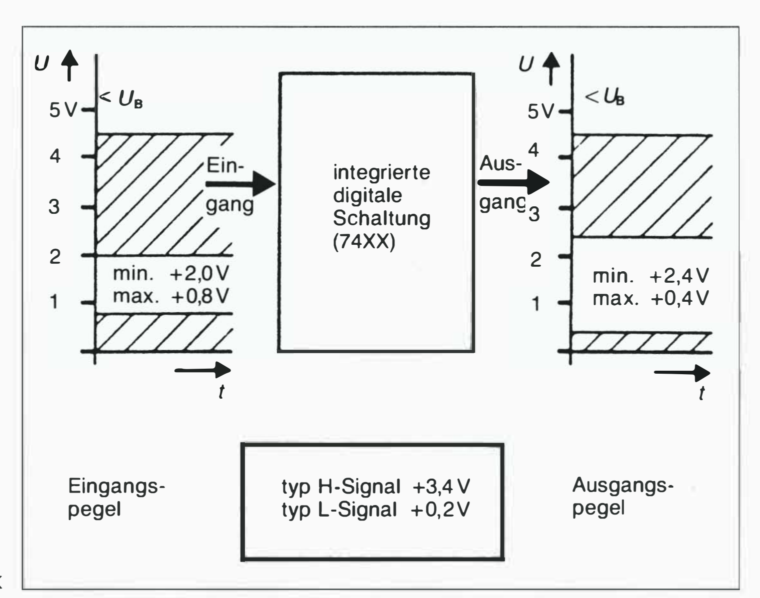


Bild 1.7: Spannungsbereiche für digitale Signale an Bausteinen der Reihe 74XX

Pegel die Rede sein, wenn Spannungen gemeint sind.
Vielleicht noch einige Worte über die Schaltskizzen. Um sie übersichtlich zu halten, wird oft die Masseverbindung nicht durchgezeichnet, sondern statt dessen ein dicker Querbalken an den entsprechenden Strich gesetzt. Alle diese Querbalken müssen in Gedanken miteinander verbunden werden.
Masse steht für das Bezugspotential in einer Schaltung, also 0 Volt. Andere Punkte haben andere Potentiale, und damit baut sich zwischen ihnen eine meßbare Spannung auf. Diese Spannung kann positiv und – wie wir noch sehen werden – auch negativ sein.
Die digitalen Bausteine der 74XX-Serie sind mit steigenden Zahlen numeriert. Die Reihe beginnt mit dem wohl universellsten Digitalbaustein überhaupt, dem Vierfach-NAND 7400. Er enthält vier NAND-Gatter mit je zwei Eingängen. Bild 1.8 zeigt seine Innenschaltung. Ähnliche NAND-Gatter gibt es mit drei, vier und sogar acht Eingängen. Natürlich passen dann weniger Gatter in ein IC-Gehäuse. Im Anhang A finden Sie eine Zusammenstellung gebräuchlicher TTL-ICs. Beachten Sie in diesem Zusammenhang dort besonders die Typen 7410, 7420 und 7430.
Die Anschlußbezeichnungen eines IC werden immer durch eine Kerbe gekennzeichnet. Legt man den Baustein so vor sich, daß er auf seinen Beinchen steht und die genannte Kerbe auf der linken Seite zu liegen kommt, so befindet sich der Anschluß 1 links unten. Alle anderen Pins werden gegen den Uhrzeigersinn durchge-

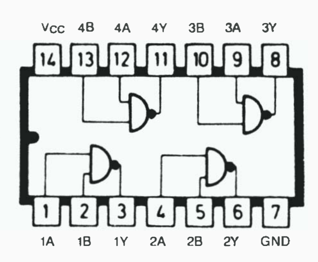


Bild 1.8: Innenschaltung eines 7400 mit vier NAND-Gattern

zählt. Genau in dieser Draufsicht sind die ICs auch in Datenblättern abgebildet.
Bei fast allen 74XX-ICs befindet sich der Anschluß für die positive Betriebsspannung (+5 Volt) links oben, der für Masse rechts unten. In Schaltplänen werden die Verbindungen zur Spannungsversorgung der Übersichtlichkeit halber oft weggelassen. Es versteht sich von selbst, daß jeder einzelne Baustein an die Betriebsspannung angeschlossen werden muß.
Grundsätzlich kann man Digitalbausteine beliebig untereinander verbinden. Nicht zulässig ist jedoch die Zusammenschaltung von zwei Ausgängen normaler TTL-Gatter nach Bild 1.9. Ein solches Gatter hat nämlich eine Innenschaltung wie in Bild 1.10 dargestellt. Einer der beiden

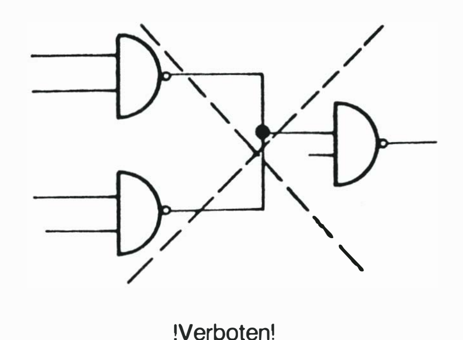


Bild 1.9: Verbotenes Zusammenschalten zweier Gatterausgänge

Transistoren Tr3 und Tr4 ist immer durchgeschaltet, der andere dagegen gesperrt. Falls beide Ausgänge verschiedene Zustände hätten, würde über kurz oder lang einer

Bild 1.10: Innenschaltung eines normalen NAND-Gatters

der Transistoren zerstört werden und das IC unbrauchbar machen. Verschiedene Gattertypen werden jedoch mit offenem Kollektor angeboten. Die zugehörige Innenschaltung zeigt Bild 1.11.
Der gestrichelt gezeichnete Kollektorwiderstand muß von außen zugeschaltet werden. Über ihn kann man so auch mehrere Gatterausgänge verknüpfen. Solchermaßen parallelgeschaltete Ausgänge ergeben immer dann eine UND-Verknüpfung, wenn das LOW- Potential dominiert (wired-AND, verdrahtetes UND) bzw. eine ODER- Verknüpfung, wenn das HIGH-Potential Vorrang hat (wired-OR, verdrahtetes ODER). Wir werden solche Schaltungen beim Busbetrieb oder bei der Interruptsteuerung noch kennenlernen.
Ein weiterer Vorteil von TTL-Bausteinen mit offenem Kollektor ist, daß über den externen Widerstand auch Verbraucher mit höheren Spannungen (meist bis zu 30 Volt) geschaltet werden können. Die Betriebsspannung des IC selbst bleibt dabei aber nach wie vor +5 Volt. Einige Anwendungen werden wir noch in Kapitel 2 kennenlernen.

1.3.3 Die verschiedenen 74XX-Familien

Eine weitere Kombinationseinschränkung ergibt sich durch die Zahl der Eingänge, die an einen Ausgang geschaltet werden dürfen. Bei Standardtypen 74XX ist sie auf maximal zehn beschränkt. Es gibt jedoch außer den Standardtypen auch noch die Familien 74LSXX, 74LXX,

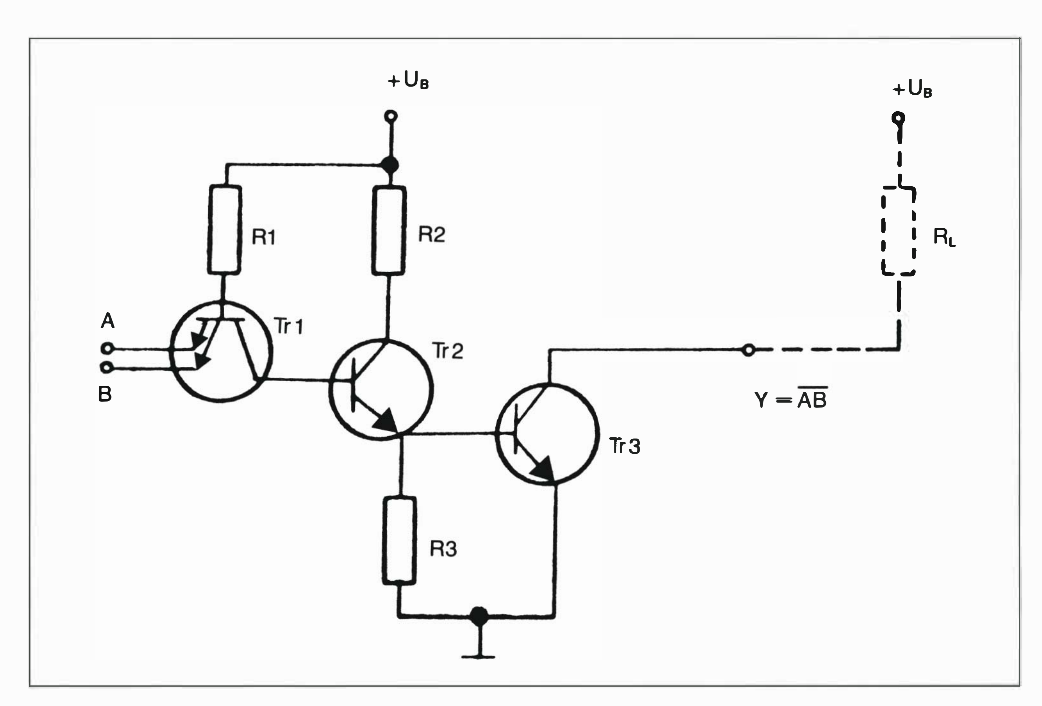


Bild 1.11: NAND-Gatter mit offenem Kollektor als Ausgang

Familie	Laufzeit/ Gatter	Leistung/ Gatter	max. Last							
			Std	ALS	AS	F	H	L	LS	S
AS	1,5ns	22mW	10	50	10	50	10	100	50	10
F	2ns	4mW	12	25	10	25	10	48	25	10
S	3ns	20mW	12	50	10	50	10	100	50	10
ALS	4ns	1mW	10	20	10	20	4	40	20	10
H	6ns	22,5mW	12	25	10	25	10	50	25	10
LS	9ns	2mW	5	20	8	50	4	40	50	10
Std	10ns	10mW	10	20	8	20	8	40	20	8
L	33ns	1mW	2	10	1	10	1	20	10	1

Tabelle 1.4: Eigenschaften der unterschiedlichen 74XX-Familien

74HXX, 74ALSXX, 74SXX, 74FXX und 74ASXX. Sie unterscheiden sich in einigen Eigenschaften. So ist ein wichtiges Kriterium die Geschwindigkeit, die jedoch meist mit anderen Nachteilen erkauft werden muß. In Tabelle 1.4 sind die unterschiedlichen TTL-Familien mit ihren wichtigsten Daten aufgelistet.

Die maximale Last gibt an, wieviele Einheiten an einen Ausgang des jeweiligen Bausteins angeschlossen werden dürfen. Zum Beispiel ist jeder Standardausgang imstande, zehn Standardeingänge zu steuern. Ein Low-Power-Baustein (L) kann dagegen nur einen einzigen Schottky-Eingang (S) treiben.

Üblich sind heute fast ausschließlich LS-Typen. Sie bieten ein Optimum zwischen Geschwindigkeit, Stromverbrauch und Preis. Falls etwas größere Lasten zu treiben sind, weicht man auf Standardtypen aus. Dies könnte beispielsweise der Fall sein, wenn entweder viele Eingänge an einem Ausgang hängen, oder wenn etwa eine Leuchtdiode direkt von einem TTL-Ausgang getrieben wird, der gleichzeitig einen weiteren Eingang ansteuert.

In zeitkritischen Fällen bieten sich die S-Typen an. Alle anderen Varianten sind im Hobbybereich nur schwer zu bekommen und dann jedenfalls sehr teuer.

In letzter Zeit tauchen am Horizont immer stärker die High-Speed- CMOS-Typen 74HCXX auf. Ihre wichtigsten Daten sind kompatibel mit den entsprechenden LS-Bausteinen, jedoch liegt ihre Stromaufnahme im Normalfall wesentlich niedriger. Vereinzelt gibt es sogar schon Bausteine, die als HC-Typ billiger sind, als in der LS-Version.

Außer den TTL-ICs gibt es noch die weit verbreitete MOS-Serie 4XXX, die sich durch ihren äußerst geringen Ruhestrombedarf auszeichnet. Sie ist bei Betriebsspannungen von 3 bis 15 Volt einsetzbar, was sie besonders für batteriegespeiste Geräte attraktiv macht. Unter folgenden Voraussetzungen können solche CMOS-Bausteine bedenkenlos zusammen mit denen der 74XX-Familien eingesetzt werden:

1. Die CMOS-Bausteine müssen ebenfalls mit einer Betriebsspannung von 5 Volt betrieben werden, und

2. ein MOS-Ausgang darf nur einen einzigen TTL-Eingang treiben.

1.3.4 Was tun mit unbenutzten Eingängen?

Bei der Konzeption digitaler Schaltungen kann es vorkommen, daß nicht alle Gattereingänge eines Logikgliedes benötigt werden. Hier muß beachtet werden, daß unbenutzte Eingänge der TTL-Bausteine in der Regel so zu bewerten sind, als seien sie mit HIGH-Potential (!) beschaltet. Dies bedeutet für NAND-Glieder keine funktionelle Einschränkung, da diese bereits mit LOW-Pegel an nur einem Eingang auf Dauer-HIGH am Ausgang gezwungen werden. Ausgänge von NOR-Gliedern jedoch werden bereits durch nur ein Eingangs-HIGH gesperrt. Zur Erzielung optimaler Schaltgeschwindigkeiten und zur Verringerung der Störempfindlichkeit sollte man aber alle Eingänge an festes Potential legen. Dies ist durch folgende Maßnahmen zu erreichen:
Grundsätzlich ist es möglich, den freien Eingang einem anderen Eingang desselben Gatters parallelzuschalten, falls der Ausgang der vorhergehenden Stufe dadurch nicht überlastet wird.
Bei NAND-Gattern kann der unbenutzte Eingang auch direkt mit der Versorgungsspannung verbunden werden, falls sichergestellt ist, daß diese +5,5 Volt nicht übersteigt. Andernfalls muß ein Vorwiderstand von etwa 1 Kiloohm vorgeschaltet werden. An diesen Widerstand können dann bis zu 25 Logik-Glied-Eingänge angeschlossen werden.
Bei NOR-Gattern kann man den freien Eingang auch direkt an Masse legen.
Die Eingänge ganzer unbenutzter Gatter und anderer Funktionsgruppen sollten auf Masse gelegt werden, um einmal eine geringere Leistungsaufnahme zu erreichen und zum anderen zu vermeiden, daß diese Elemente durch Störsignale zum Schalten veranlaßt werden, was unnötige Ströme auf den Versorgungsleitungen zur Folge hätte.
Oft sind einzelne IC-Anschlußbeinchen mit N.C. bezeichnet. Diese Abkürzung steht für Not Connected (nicht verbunden). Es ist jedoch trotzdem möglich, daß bei manchen Herstellern hier eine Verbindung zu Testzwecken herausgeführt wurde. Man sollte daher vorsichtshalber solche Anschlüsse grundsätzlich nicht beschalten.
Um Störungen abzublocken, ist es sinnvoll, sogenannte Koppelkondensatoren von etwa 0,1 μF dicht neben jedem IC zwischen +5 Volt und Masse zu schalten. Diese Kondensatoren sollten vom Typ Keramik bzw. Tantal sein. Sie sind dann nämlich in der Lage, die bei Schaltvorgängen auftretenden Betriebsspannungseinbrüche kurzfristig auszugleichen.

1.4 Schaltungen mit Gedächtnis

1.4.1 Flipflops

Am Anfang dieses Kapitels wurden Lichtschalter, Klingelknopf und Kugelschreiber als Beispiele für digitale Systeme genannt. Diese Dinge können in zwei Gruppen eingeteilt werden. Der Klingelknopf gehört zur ersten Gruppe. Die Klingel läutet nur solange, wie der Knopf gedrückt ist. Wird er losgelassen, dann hört die Klingel auf zu läuten, und man sieht dem Knopf nicht

an, ob er jemals betätigt worden ist. Die Gegenstände der Gruppe 2 jedoch haben ein »Gedächtnis«. Wenn man den Schalter betätigt, bleibt das Licht solange eingeschaltet, bis er erneut betätigt wird. Ebenso bleibt die Kugelschreibermine solange im eingezogenen Zustand, bis sie herausgedrückt wird. Anschließend verbleibt sie in dem neuen Zustand.

Diese Eigenschaft kann man mit einem einzigen IC 7400 nachbilden. Wir schauen uns dazu die rückgekoppelte Schaltung nach Bild 1.12 an. Rückgekoppelt heißt sie, weil zwei Leitungen vom Inneren der Schaltung wieder auf zwei Eingänge zurückgeführt werden. Diese Rückführung ist auch verantwortlich für ihr Speicherverhalten.

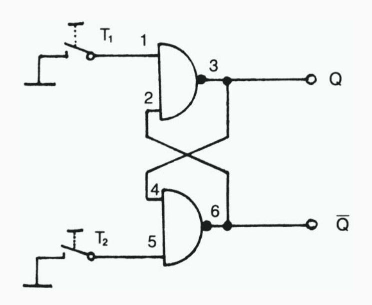


Bild 1.12: RS-Flipflop aus NANDs 74LS00

Überlegen wir uns einmal theoretisch, was in der Schaltung vorgeht. In Gedanken schließen wir den Taster T_1 und legen damit LOW an den entsprechenden Eingang. Unabhängig davon, welche logische Spannung am anderen Eingang des NAND-Gatters liegt, wird Q sofort auf HIGH gehen. Das können wir aus Tabelle 1.3 ablesen. Der erste Eingang des zweiten NAND-Gatters ist mit Q verbunden und also ebenfalls HIGH. Der zweite Eingang bleibt unbeschaltet, und damit (wie wir schon weiter oben gesehen haben) sowieso HIGH. Somit wird $\overline{Q}$ LOW.

Dieser Zustand bleibt auch erhalten, wenn T_1 wieder geöffnet wird. Der LOW-Zustand an $\overline{Q}$ zwingt Q, im HIGH-Zustand zu bleiben. Der andere Ausgang verharrt dagegen auf LOW. Man kann sagen, Q ist gesetzt worden.

Schließt man nun T_2, so wird die Sache umgekehrt. Unabhängig davon, welchen Zustand Q hatte, geht $\overline{Q}$ auf HIGH, Q dagegen wird auf LOW gezwungen. Man kann auch sagen, Q wird rückgesetzt. Auch dieser Zustand bleibt stabil, wenn beide Taster wieder geöffnet sind.

Die Schaltung hat also zwei stabile Zustände. Sie kippt entweder in den einen oder in den anderen. Daher wird sie lautmalerisch auch Flipflop genannt, eine Bezeichnung, die offensichtlich noch aus der Zeit der Relaisschaltungen stammt. In diesem Fall handelt es sich um ein Setz-Rücksetz-Flipflop.

1.4.2 Ein Flipflop als prellfreier Taster

Bei der Betätigung einfacher Taster kann es vorkommen, daß die Kontakte prellen. Dann entsteht eine Signalabgabe nach Bild 1.13. Hier wurde die Spannung in Abhängigkeit von der Zeit (t) in einem sogenannten Zeit- oder Impulsdiagramm aufgetragen. Dabei schreitet die Zeit von links nach rechts fort. Man sieht, daß die Spannung sprunghaft von LOW nach HIGH ansteigt und dort bleibt, bis der Taster losgelassen wird. Den realen Spannungsverlauf zeigt das untere Diagramm: Es entstehen einige Spannungssprünge, bis der erwartete Wert dauerhaft eingenommen wird. Obwohl man nur einen Im-

A H L ohne Kontaktprellen t

A H L mit Kontaktprellen t

+5V Rv A T Versuchsschaltung

Bild 1.13: Signalabgabe eines Tasters mit und ohne Kontaktprellen

puls abgeben wollte, sind daraus (unkontrolliert) gleich mehrere geworden. Das liegt an den mechanischen Eigenschaften der Kontaktfahnen. Natürlich tritt das Kontaktprellen während einer sehr sehr kurzen Zeit auf und ist sicher mit bloßem Auge nicht erkennbar. Steuert man über derartige nichtentprellte Taster jedoch elektronische Zähler an, so machen diese ungewollt gleich mehrere Zählschritte. Abhilfe schaffen hier elektronische Zusätze zum Entprellen der Taster. Bild 1.14 zeigt die für diesen Zweck leicht abgeänderte und etwas erweiterte Flipflop-Schaltung.

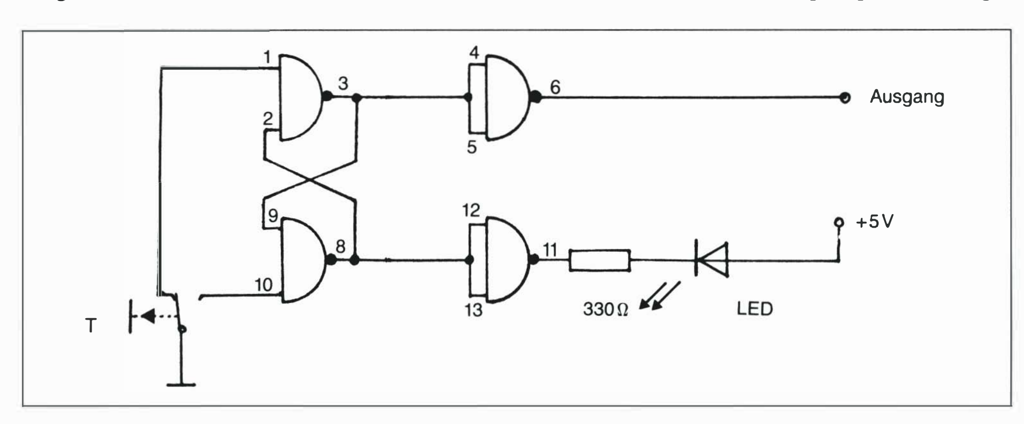


Bild 1.14: Entprellung eines Tasters zur Signaleingabe

Ausgang A führt in Ruhestellung des Tasters T LOW-Pegel. Bereits nach dem ersten Berühren des Kontaktstückes im Inneren des Tasters steuert das Flipflop schnell und sicher den Ausgang A auf HIGH-Pegel. Dies wird durch die Leuchtdiode angezeigt. Gibt man den Taster wieder frei, so wird beim ersten Berühren des Ruhekontaktes auch das Flipflop zurückgestellt und der Ausgang A führt wieder LOW-Signal.

1.5 Die erste Platine

Den gerade beschriebenen entprellten Taster wollen wir einmal aufbauen. Er wird uns bei späteren Experimenten – zum Beispiel mit den Timern und an der seriellen Schnittstelle des C64 – noch wertvolle Dienste leisten.

Bild 1.15 zeigt das Platinenlayout. Bei dieser Gelegenheit sei bereits auf Anhang A hingewiesen, in dem genaue Einzelheiten zur Platinenfertigung, zur Bestückung und zu den wichtigsten Bauteilen beschrieben sind. Haben Sie bisher noch nichts mit elektronischen Schaltungen zu tun gehabt, dann lesen Sie sich diesen Anhang zunächst bitte eingehend durch. Manche Bauteile sind sehr empfindlich und gehen sofort in die ewigen Jagdgründe der Elektronik ein, wenn man sie etwa verpolt, beim Einlöten zu heiß werden läßt, oder wenn einzelne Füßchen eines IC berührt werden.

Doch keine Angst, bei dieser kleinen Schaltung kann noch nicht viel schiefgehen. Die wenigen Teile sind in der Tabelle 1.5 festgehalten. Für die Lötnägel bohren Sie bitte etwas größere Löcher als für die übrigen Teile, dennoch auf guten Sitz achten, damit sie beim Löten nicht herausfallen. Achten Sie bei der Bestückung nach Bild 1.16 besonders auf die richtige Lage der Leuchtdiode und des IC (Kerbe!). Beide sind aus dem Bestückungsplan ersichtlich. Der Taster wird so eingebaut, daß seine bewegliche Kappe zum IC hin ansteigt. Es empfiehlt sich immer, die ICs auf Sockel zu setzen. So auch den hier verwendeten Baustein 74LS00. Er ist dann bei Verdacht auf einen Defekt leicht austauschbar und kann bei akuter Teile-

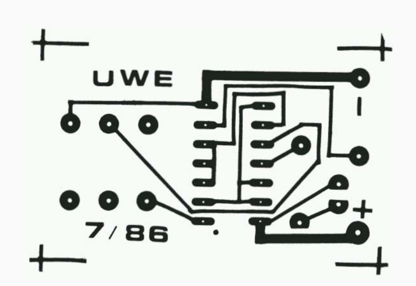


Bild 1.15: Layout des entprellten Tasters

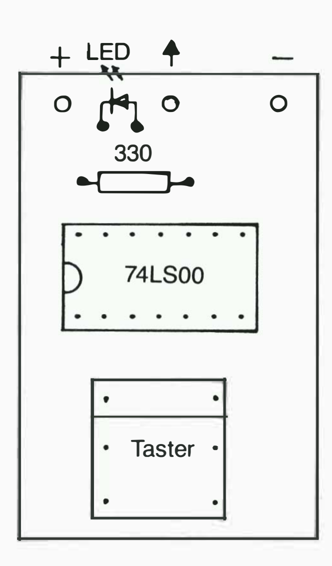


Bild 1.16: Bestückungsplan des entprellten Tasters

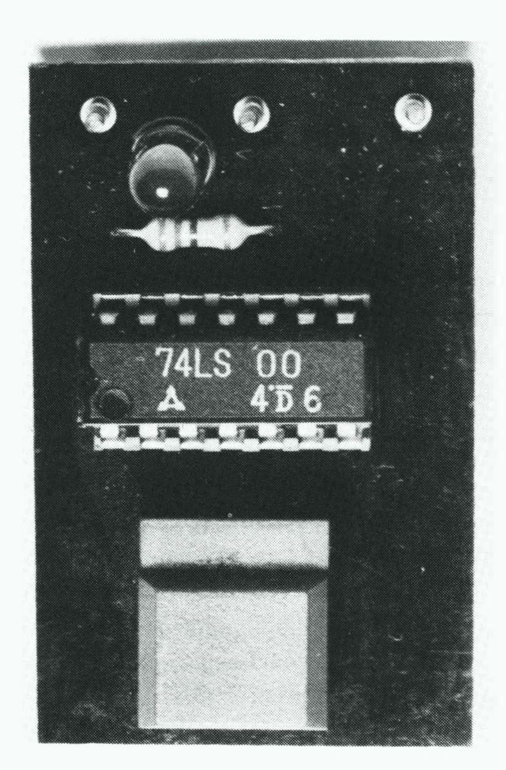


Foto 1.1: Die fertige Platine mit entprelltem Taster

knappheit einfach in eine andere Schaltung gesteckt werden.

Aus Foto 1.1 können Sie ersehen, wie die bestückte Platine aussieht. Natürlich müssen Sie später bei der Inbetriebnahme auch die Betriebsspannung richtig anlegen. Die Anschlußpunkte sind auf der Lötseite mit + und – gekennzeichnet.

1 IC 74LS00
1 IC-Sockel 14-pol
1 Leuchtdiode (Farbe nach Wahl)
1 Widerstand 330 Ohm
1 Taster 1 mal um »Digitast-mini«
3 Lötnägel
1 einseitige Platine nach Bild 1.15

Tabelle 1.5: Der Einkaufszettel für den entprellten Taster

Apropos Spannung – zum Ausprobieren der kleinen Schaltung reicht eine neue 4,5-Volt-Flachbatterie vollkommen aus. Im unverbrauchten Zustand gibt sie bei dem geringen Strombedarf durchaus noch die benötigte Spannung ab. Weiter unten werden wir dann unter anderem die Schaltung eines kräftigen Netzgerätes kennenlernen, das Batterien auf Dauer ersetzen kann.

1.6 Das D-Flipflop – ein Beispiel für TTL-Spezialschaltungen

Wie schon erwähnt, kann man allein unter Verwendung der logischen Grundbausteine UND und NICHT jedes beliebige digitale Gerät aufbauen. Versucht man das aber, so macht man es sich selbst unnötig schwer, denn der Aufwand an Grundgattern steigt schnell an und man würde sehr leicht den Überblick verlieren. Bild 1.17, Seite 30, zeigt ein aus drei einzelnen 74LS10 (je zwei NAND-Gatter mit drei Eingängen) zusammengesetztes D-Flipflop. Zudem läßt sich feststellen, daß bestimmte Elemente immer wieder in der gleichen Form gebraucht werden. Das haben auch die Hersteller von TTL-Bausteinen früh erkannt. Das kompliziert aussehende D-Flipflop aus Bild 1.17 dient beispielsweise als Register zum Speichern von Daten. Der am Eingang D liegende Zustand wird bei einem HIGH-LOW-Übergang am Takteingang C an den Ausgang Q übernommen. Er bleibt dort erhalten, wenn sich die Spannung an D auch ändert, bis wieder eine negative Flanke an C auftritt.

D-Flipflops finden oft Verwendung, und auch wir werden sie später noch genauer

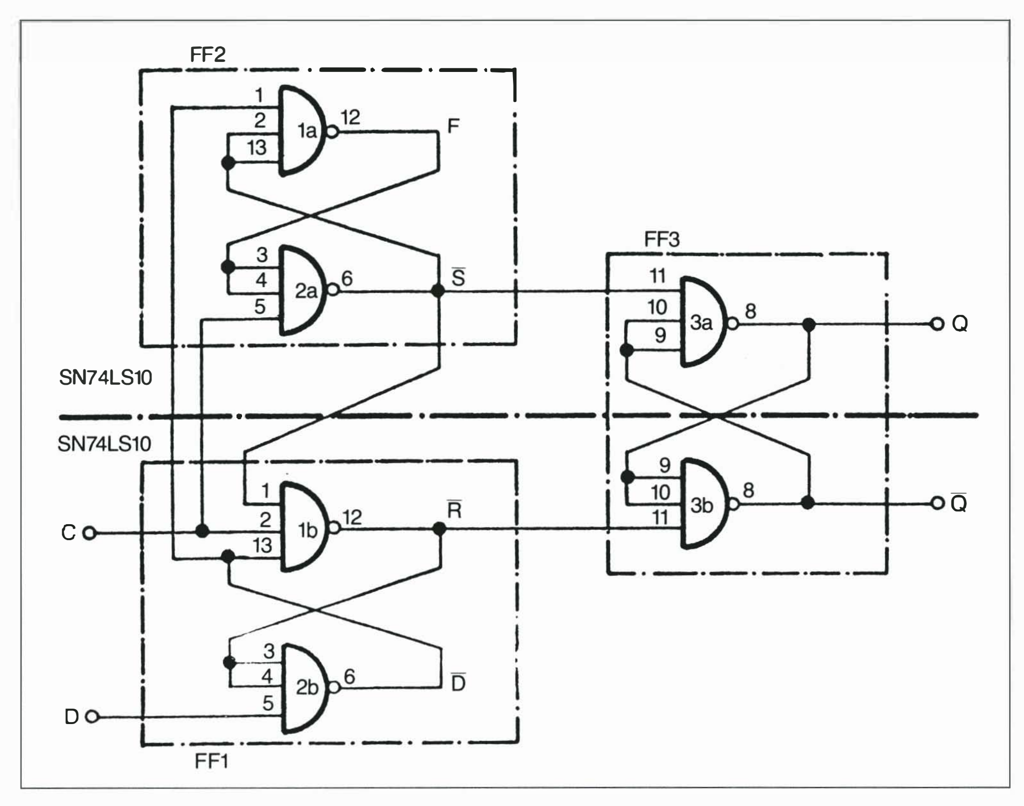


Bild 1.17: Eine Flipflopschaltung aus Einzelgattern

kennenlernen. Sie werden daher komplett als 74LS74 geliefert. Dort befinden sich sogar gleich zwei Flipflops in einem Gehäuse, so daß die Schaltung wesentlich bequemer aufzubauen, platz- und auch noch stromsparender ist.
Natürlich kann man nicht in jeden Schaltplan die gesamte Innenschaltung mit einzelnen Gattern eintragen. Man verwendet daher bestimmte Symbole, die eindeutig die Funktion des jeweiligen Elements festlegen. Auch die Logiksymbole für UND, ODER und NICHT sind ja bereits abstrahierte Zeichen. Nur so sind auch größere Schaltungen leicht durchschaubar. Kein Mensch käme auf die Idee, für jedes einzelne Gatter Transistorschaltungen wie etwa in Bild 1.10 zu verwenden.
Im nächsten Abschnitt werden wir bereits einige dieser spezielleren Bausteine kennenlernen und auch eine nützliche Schaltung damit aufbauen.

1.7 Zählerschaltungen

1.7.1 Das Toggle-Flipflop

Gleich zu Anfang dieses Kapitels wurde ein Kugelschreiber als Beispiel für digitale Speicherglieder genannt. Drückt man den Knopf ein und läßt ihn wieder los, so wird die Mine herausgeschoben und bleibt auch in dieser Stellung. Erst wenn man

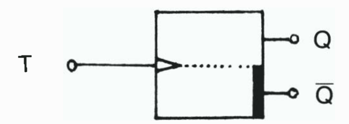


Bild 1.18: Symbol des Toggle-Flipflops mit Triggerung durch positive Flanken

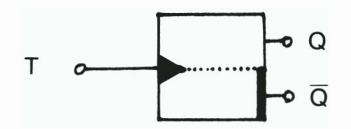


Bild 1.19: Symbol des Toggle-Flipflops mit Triggerung durch negative Flanken

noch einmal drückt und wieder losläßt, springt sie in ihre ursprüngliche Lage. Das Drücken auf den Knopf können wir als Eingangsimpuls bezeichnen, den Ausgangszustand bildet die Lage der Mine.
Ganz anolog dazu verhält sich das sogenannte Toggle-Flipflop. Bild 1.18 gibt das zugehörige Symbol wieder. Man erkennt den Takteingang T und die zwei komplementären Ausgänge Q und $\overline{Q}$. Komplementär sagt man, weil diese Ausgänge immer entgegengesetzte Signale führen. Liegt Q auf HIGH, befindet sich $\overline{Q}$ immer auf LOW und umgekehrt.
Toggle heißt einfach wechseln. Wie bei dem Kugelschreiber genügt ein Signal an T, um beide Ausgangszustände umzukehren. Natürlich wird es nichts helfen, auf den Eingangsanschluß zu drücken, sondern man verwendet einen Spannungsimpuls. Das in Bild 1.18 dargestellte Flipflop ist vorderflankengesteuert, das heißt, es reagiert ausschließlich auf die positive Flanke des Eingangsimpulses, also auf den Wechsel von LOW nach HIGH. Kenntlich ist das an der nicht ausgefüllten Spitze am Takteingang.
Bild 1.19 zeigt dagegen ein rückflankengesteuertes Flipflop. Hier ist die Spitze ausgefüllt. Daher reagiert diese Variante ausschließlich auf negative Taktflanken, also auf HIGH-LOW-Übergänge. Das

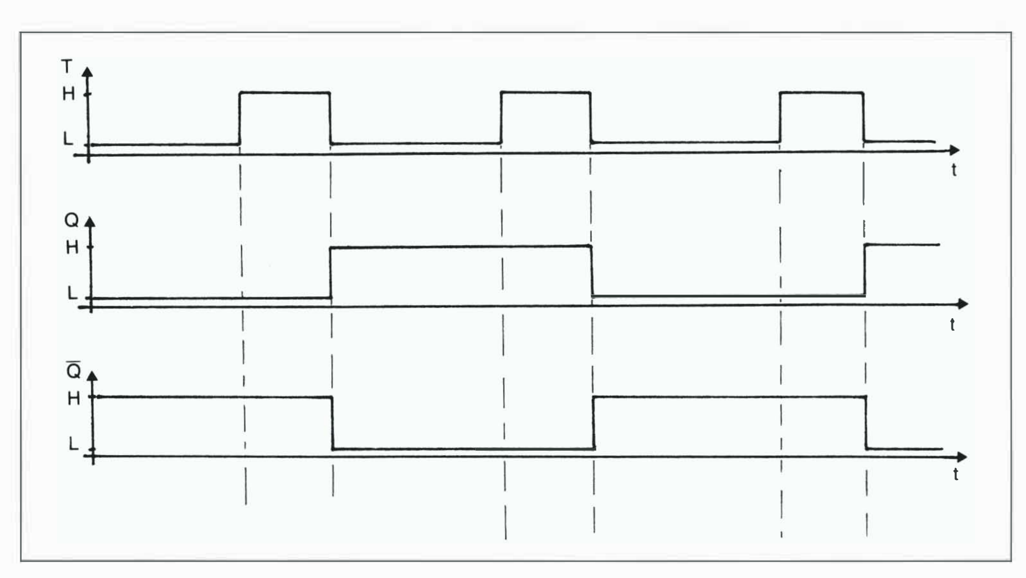


Bild 1.20: Zeitdiagramm eines negativ flankengetriggerten Toggle-Flipflops

Zeitdiagramm in Bild 1.20 macht die Arbeitsweise dieses letzten Flipflops noch einmal grafisch deutlich.

In der Praxis werden beide Typen nebeneinander eingesetzt. Leider ist die Kennzeichnung uneinheitlich und nicht immer eindeutig. Es empfiehlt sich also, das zugehörige Datenblatt genau zu studieren.

Nehmen wir an, daß der Ausgang Q zunächst auf LOW-Pegel liegt. Der komplementäre Ausgang $\overline{Q}$ befindet sich dann natürlich genau entgegengesetzt auf HIGH. In der obersten Zeile ist der Zustand des Takteingangs T aufgetragen, nach rechts schreitet die Zeit (t) fort. Wir sehen, daß die Spannung an diesem Eingang zunächst ebenfalls LOW ist und nach einer Weile auf HIGH-Potential ansteigt. Da wir ein rückflankengesteuertes Flipflop betrachten, tut sich am Ausgang noch nichts. Bald darauf aber fällt die Eingangsspannung auf LOW ab. Jetzt ist die Schaltbedingung für das Flipflop erfüllt, es ist nämlich eine negative Flanke am Takteingang aufgetreten. Die Ausgangszustände werden also gewechselt: Q springt auf HIGH und $\overline{Q}$ auf LOW. Man sagt auch, das Flipflop wurde getriggert.

Wieder einige Zeit später geht T erneut auf HIGH. Auch jetzt passiert natürlich nichts. Erst beim erneuten Wechsel der Eingangsspannung zurück auf LOW kippt die Ausgangsstufe wieder und wandelt Q in LOW und $\overline{Q}$ in HIGH um. Dies wiederholt sich zyklisch bei jedem Taktimpuls.

1.7.2 Zwei Kulis können bis drei zählen !

Eine der wichtigsten Anordnungen in der Digitaltechnik sind Zähler. Nachdem Sie nun wissen, wie ein Toggle-Flipflop funktioniert, ist es zum Aufbau solcher Schaltungen nur noch ein kleiner Schritt. Wieder einmal muß der Kugelschreiber als Beispiel herhalten. Nehmen Sie an, Sie hätten zwei Kulis, deren Mine erst beim Loslassen des Druckknopfes ihre Lage änderte, und zwar ausreichend weit, um ihrerseits wieder einen anderen Kuliknopf bedienen zu können. Diese beiden Kugelschreiber legen Sie jetzt mit eingezogenen Minen so hintereinander, wie in Bild 1.21 zu sehen ist.

Der Zustand der Minen soll der Zählstand sein. Zu Anfang sind beide eingezogen. Es wurde noch nicht gedrückt. Dieser Zustand entspricht also der Null. Nun drücken wir einmal auf den Knopf des rechten Kulis und lassen wieder los. Die rechte Mine wird herausspringen, die linke jedoch eingezogen bleiben. Das ist also die Zählerstellung 1. Drücken wir noch einmal, so wird die Mine des rechten Kulis wieder zurückspringen, dafür aber beim Loslassen des linken Kuliknopfes dessen Mine herausspringen lassen. Damit haben wir die Zählerstellung 2. Beim dritten Eingangsimpuls am rechten Kuli wird nur dessen Mine herausspringen. Beide Minen ausgeschoben bedeutet also 3. Der

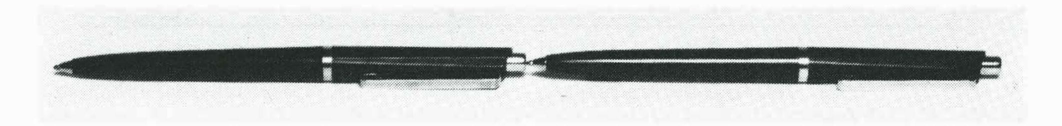


Bild 1.21: Ein Zähler aus Kugelschreibern

folgende Impuls führt nun dazu, daß beide Minen wieder in ihre Ausgangsstellung zurückspringen. Diese Kombination hatten wir bereits mit der Zahl 0 bezeichnet. Ab nun wiederholen sich bei erneuten Impulsen alle vier Stellungen immer wieder. Fassen wir das Ergebnis noch einmal in Tabelle 1.6 zusammen. Für die beiden Lagen der Mine setzen wir jeweils eine 0 für eingezogen und eine 1 für ausgeschoben. Damit erhalten wir folgendes Bild:

Probieren Sie das ruhig einmal bei unserem Kulizähler. Der rechte Stift hat die Wertigkeit 1, der linke 2. Ragt beispielsweise nur die linke Mine heraus, so ist der dargestellte Wert 2, ragen beide heraus, dann ergibt sich 2+1=3.

Anhang E enthält eine Tabelle, in der alle 256 mit 8 Bit erreichbaren Kombinationen jeweils mit ihrem dezimalen und hexadezimalen Wert aufgeführt sind.

linke Mine	rechte Mine	dargestellte Zahl
0	0	0
0	1	1
1	0	2
1	1	3

Tabelle 1.6: Zustände eines Zählers mit 4 Stellungen

So kommen wir zu dem Schluß, daß zwei Kulis tatsächlich nur bis drei zählen können. Erweitern wir die Zählkette aber nach links, dann können wir auch größere Zahlen darstellen. Die Darstellungsweise ist hier das Dualsystem. Jede Stelle hat dabei eine bestimmte Wertigkeit. Die niederwertigste zum Beispiel 1, die nächsthöhere 2, die folgende 4, 8, 16, 32 und so weiter. Ganz allgemein kann man die dargestellte Dezimalzahl errechnen, indem man die Wertigkeiten der Stellen addiert, an denen die Dualzahl eine 1 hat. Alle Stellen mit 0 können unter den Tisch fallen.

1.7.3 Digitale Zähler aus Flipflops

Genauso wie unser Zähler aus Kugelschreibern, arbeiten auch elektronische Digitalzähler. Natürlich werden hier statt der Kulis Toggle-Flipflops eingesetzt. Bild 1.22 zeigt den Aufbau eines vierstufigen Zählers. Er hat vier Ausgänge Q_1 bis Q_4. Die höchste Zahl, die mit ihm dargestellt werden kann, ist also 1+2+4+8=15. Die komplementären Ausgänge $\overline{Q_1}$ bis $\overline{Q_4}$ sind nicht beschaltet und wurden daher gleich weggelassen.

Bild 1.23 enthält das zugehörige Impuls-

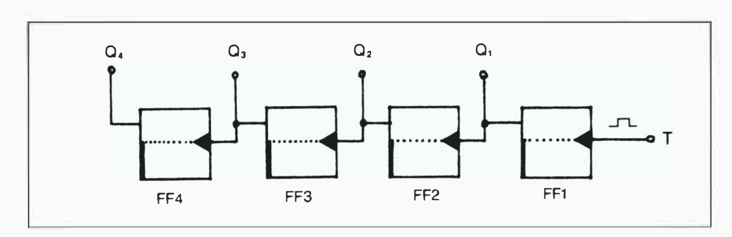


Bild 1.22:
Vierstelliger Zähler aus Flipflops

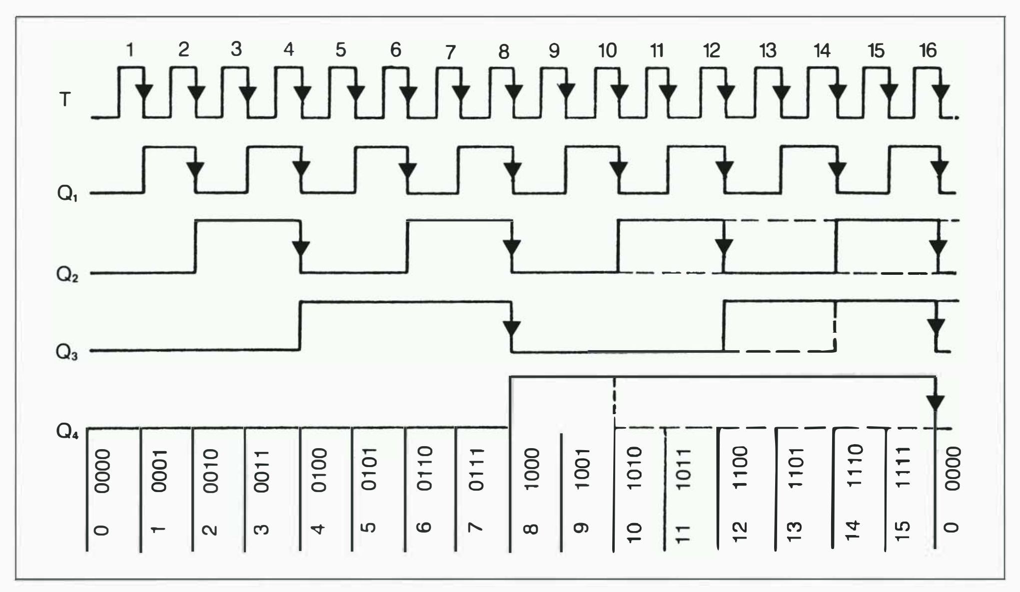


Bild 1.23: Impulsdiagramm eines vierstelligen Zählers, gestrichelt die Zustände eines BCD-Zählers

diagramm. Man sieht deutlich, daß sich der Zustand der nachfolgenden Stufen erst bei einer fallenden Flanke am Eingang ändert.

Wie bereits gesagt, kann die Schaltung aus Bild 1.22 von 0 bis 15 zählen, also 16 Zustände haben. Sie ist damit ein Hexadezimalzähler, das heißt, sie durchläuft alle Ziffern des Zahlensystems zur Basis 16. Wie Sie wahrscheinlich bereits wissen, werden die 16 Zeichen dort durch die Dezimalziffern 0 bis 9 und zusätzlich die Buchstaben A bis F bezeichnet. Tabelle 1.7 enthält eine übersichtliche Zusammenstellung der verschiedenen Systeme.

1.7.4 Menschenfreundliche Elektronik zählt bis zehn

Der Mensch ist aus anatomischen Gründen von jeher an das Dezimalsystem gewöhnt. Er rechnet mit Zahlen, in denen nur Ziffern von 0 bis 9 vorkommen. Will man also einen menschenfreundlichen Zähler aufbauen, dann sollte man dafür sorgen, daß Ergebnisse mit Hexadezimalziffern über 9 nicht vorkommen können. Sobald die Ziffer A, also die Stellung 1010 auftauchen will, sollte man den Zähler bereits auf 0000 zurückstellen und die nächste Stelle anbrechen.

Dazu gibt es an vielen Flipflops einen Rückstell- oder Reset-Eingang, der bei LOW-Spannung sofort den Normalzustand (Q = LOW, $\overline{Q}$ = HIGH) einstellt, unabhängig davon, was der Takt gerade macht. In Tabelle 1.7 sehen wir, daß bei der betrachteten Zahl 1010 zum ersten Mal Q_4 UND Q_2 gleichzeitig beide HIGH sind. Daraus können wir bereits die Rücksetzbedingung ableiten.

Bild 1.24 zeigt die nötige Schaltung. Jetzt können nur noch die Zustände 0 bis 9 erreicht werden. Die Hexadezimalziffer A

Wertigkeit:	Dual 8 4 2 1	Dezimal 10 1	Hexadezimal 16
	0 0 0 0	0	0
	0 0 0 1	1	1
	0 0 1 0	2	2
	0 0 1 1	3	3
	0 1 0 0	4	4
	0 1 0 1	5	5
	0 1 1 0	6	6
	0 1 1 1	7	7
	1 0 0 0	8	8
	1 0 0 1	9	9
	1 0 1 0	1 0	A
	1 0 1 1	1 1	B
	1 1 0 0	1 2	C
	1 1 0 1	1 3	D
	1 1 1 0	1 4	E
	1 1 1 1	1 5	F

Tabelle 1.7: Die Zählerstände in unterschiedlichen Darstellungsweisen

liegt nur sehr kurze Zeit an, da der Zähler sofort wieder auf 0 gestellt wird. Dieser Zustand ist also nicht stabil.
An den vier Ausgängen stehen damit die Dezimalziffern binär kodiert zur Verfügung. Man sagt daher auch BCD-Zähler zu der Anordnung für binär kodierte Dezimalzahlen, oder englisch Binary Coded Decimals. In Bild 1.23 ist der Spannungsverlauf für BCD- Zähler gestrichelt eingetragen.
Die TTL-Serie 74XX verfügt sogar über mehrere Bausteine, die intern bereits einen kompletten BCD-Zähler enthalten. Einer dieser Bausteine ist der 7490, den wir im Folgenden verwenden wollen. Seine Innenschaltung ist symbolisch in Bild 1.25 dargestellt.

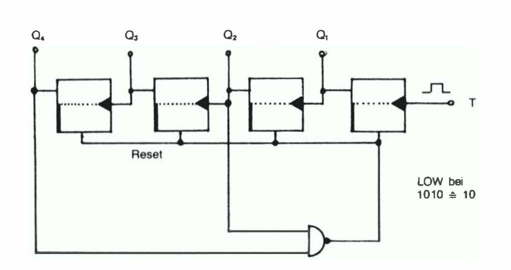


Bild 1.24: Dezimalzähler durch Rückstellung bei 1010

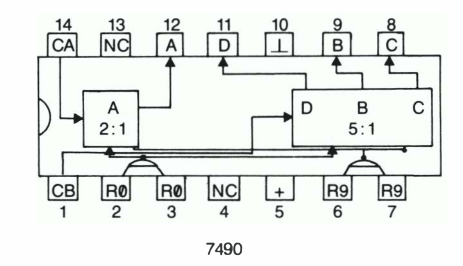


Bild 1.25: Pinbelegung des 7490

1.7.5 Das Ableseproblem

Den menschenfreundlichen Zähler haben wir nun, doch was nutzen uns irgendwelche Spannungen an vier Ausgängen? Man könnte an jeden Ausgang eine Leuchtdiode anschließen, doch wer möchte schon dauernd im Kopf Wertigkeiten addieren?

Aber keine Angst, auch hier weiß die 74XX-Reihe zu helfen. Alle Dezimalziffern sind gut lesbar mit einer Sieben-Segment-Anzeige darzustellen. Diese Anzeigeart kennen Sie alle von Taschenrechnern und Digitaluhren. Dabei sind einfach sieben balkenförmige Leuchtdioden in Form einer stilisierten 8 angeordnet. Je nachdem, welche Balken leuchten, ergibt sich eine Ziffer in Klarschrift.

Die Sache hat nur einen Haken. Wie macht man aus vier Zählerausgängen sieben LED-Ansteuerungen? Nun – man nehme einen 7447. Dieser TTL-Baustein besorgt die Umwandlung. Er hat vier Eingänge und sieben Ausgänge, die er nach Bild 1.26 zuordnet.

Dieses Bild zeigt auch die Anschlußbelegung des BCD zum Sieben-Segment-Umsetzer. Die vier Eingänge sind mit A, B, C und D bezeichnet, die Ausgänge mit den kleinen Buchstaben $\overline{a}$ bis $\overline{f}$. Hier wird der Anzeigebaustein angeschlossen.

Beim Betrieb müssen – wie immer bei LEDs – Strombegrenzungswiderstände eingefügt werden. Ihr Wert richtet sich nach der gewünschten Helligkeit der Anzeige und beträgt etwa 330 Ohm. Im Normalbetrieb liegen die Anschlüsse $\overline{LT}$ (Lamp Test, Pin 3) und $\overline{BI/RBO}$ (Ripple Blanking Output, Pin 4) auf HIGH. Der Zustand von $\overline{RBI}$ (Ripple Blanking Input, Pin 5) ist beliebig.

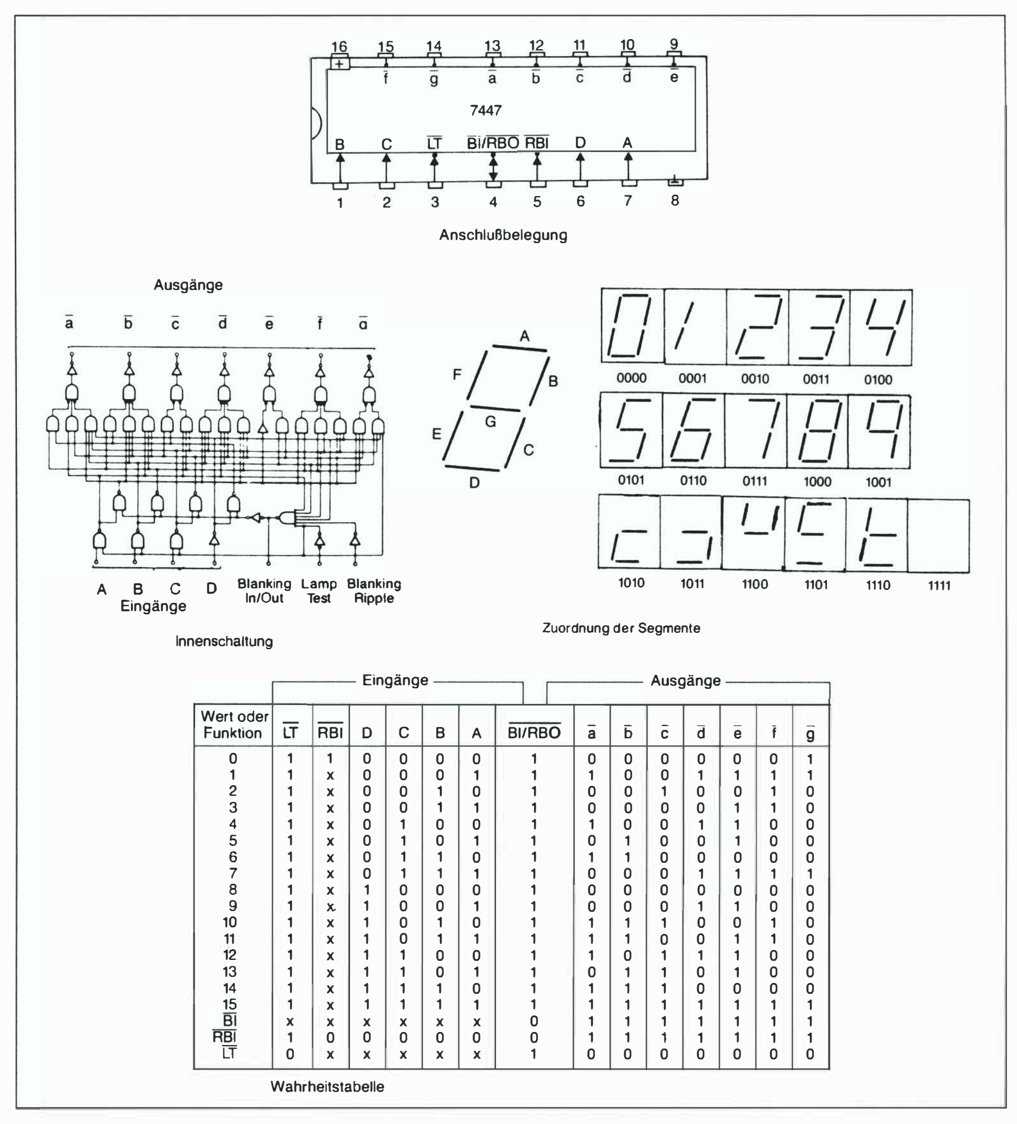
Der Vollständigkeit halber sei hier noch folgendes zur Funktion dieser Pins angemerkt:

Eine Überprüfung der Anzeige kann erfolgen, indem man $\overline{LT}$ auf LOW legt. In diesem Fall müssen alle Segmente aufleuchten.

Eine Unterdrückung führender Nullen in mehrstelligen Anzeigen erhält man, indem der Ausgang $\overline{BI/RBO}$ einer Stelle mit dem Eingang $\overline{RBI}$ der nächstniedrigeren Stufe verbunden wird. $\overline{RBI}$ der höchstwertigen Stufe sollte hierbei an Masse gelegt werden. Die Anzeigen vor der ersten gültigen Ziffer bleiben dann dunkel. Da im allgemeinen eine Nullenunterdrückung in der niedrigstwertigen Stufe nicht gewünscht wird, läßt man $\overline{RBI}$ dieser Stufe offen. Ähnlich kann man nachlaufende Nullen in gebrochenen Dezimalzahlen unterdrücken.

In der Anzeige selbst wurden bereits zur Einsparung von Anschlußbeinchen alle sieben Leuchtbalken mit jeweils einem Pol an einen einzigen gemeinsamen Pin gelegt. Der Querstrich über den Ausgangsbezeichnungen des Dekoders deutet ja schon an, daß die Ausgangsspannung für einen leuchtenden Balken LOW ist. Man sagt auch, die Ausgänge sind Aktiv-LOW. Für den Baustein 7447 muß also eine Anzeige gewählt werden, bei der die Pluspole (Anoden) aller sieben Segmente zusammengeschaltet sind. Es gibt auch Anzeigen mit gemeinsamem Minuspol (Kathode). Diese benötigen jedoch einen anderen Dekoder.

Mit den gesammelten Kenntnissen sind wir bereits in der Lage, einen kompletten einstelligen Dezimalzähler mitsamt Anzeige aufzubauen. Bild 1.27, Seite 38, zeigt



	Eingänge							Ausgänge						
Wert oder Funktion	$\overline{LT}$	$\overline{RBI}$	D	C	B	A	$\overline{BI/RBO}$	$\bar{a}$	$\bar{b}$	$\bar{c}$	$\bar{d}$	$\bar{e}$	$\bar{f}$	$\bar{g}$
0	1	1	0	0	0	0	1	0	0	0	0	0	0	1
1	1	x	0	0	0	1	1	1	0	0	1	1	1	1
2	1	x	0	0	1	0	1	0	0	1	0	0	1	0
3	1	x	0	0	1	1	1	0	0	0	0	1	1	0
4	1	x	0	1	0	0	1	1	0	0	1	1	0	0
5	1	x	0	1	0	1	1	0	1	0	0	1	0	0
6	1	x	0	1	1	0	1	1	1	0	0	0	0	0
7	1	x	0	1	1	1	1	0	0	0	1	1	1	1
8	1	x	1	0	0	0	1	0	0	0	0	0	0	0
9	1	x	1	0	0	1	1	0	0	0	1	1	0	0
10	1	x	1	0	1	0	1	1	1	1	0	0	1	0
11	1	x	1	0	1	1	1	1	1	0	0	1	1	0
12	1	x	1	1	0	0	1	1	0	1	1	1	0	0
13	1	x	1	1	0	1	1	0	1	1	0	1	0	0
14	1	x	1	1	1	0	1	1	1	1	0	0	0	0
15	1	x	1	1	1	1	1	1	1	1	1	1	1	1
$\overline{BI}$	x	x	x	x	x	x	0	1	1	1	1	1	1	1
$\overline{RBI}$	1	0	0	0	0	0	0	1	1	1	1	1	1	1
$\overline{LT}$	0	x	x	x	x	x	1	0	0	0	0	0	0	0

Wahrheitstabelle

Bild 1.26: Daten des Dekodierers/Treibers 7447

die entsprechende Schaltung. Natürlich kann dieser Zähler nur von 0 bis 9 laufen. Will man den Zählbereich erweitern, dann benötigt man eine zusätzliche Stelle. Dazu nimmt man einfach die gleiche Einheit noch einmal und benutzt als Eingangssignal die logische Spannung am Ausgang Q_4 der vorhandenen Stufe. Beim Wechsel von 9 (1001) nach 0 (0000) tritt ja hier eine negative Flanke auf, die das nachfolgende Flipflop triggern kann.

In Bild 1.28 ist die entsprechende Anord-

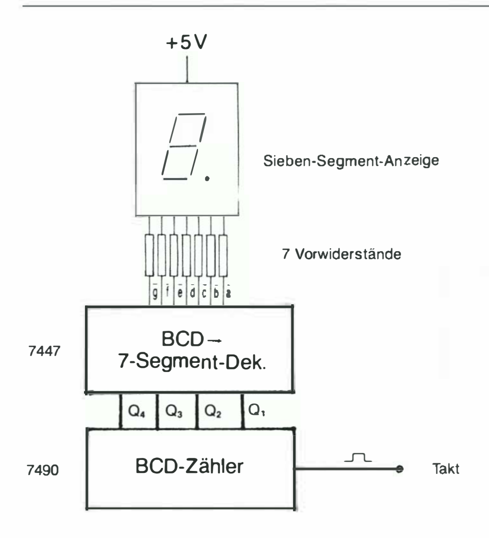


Bild 1.27: Zähler mit Anzeige (0 bis 9)

nung zu sehen. Der Zählbereich geht jetzt von 00 bis 99.
Zusätzlich ist noch eine Reset-Taste angebracht. Der BCD-Zähler hat – wie bereits erwähnt – eine Rückstellmöglichkeit. Für normalen Zählbetrieb muß mindestens einer der beiden Reset-Eingänge R Ø auf LOW liegen. Falls beide Eingänge zugleich HIGH gemacht werden, setzt der 7490 alle seine Ausgänge auf LOW zurück.
Das wird in der Schaltung durch einen Taster mit Pull-Up-Widerstand erreicht. Falls der Taster geschlossen ist, liegt die Reset-Leitung direkt auf Masse-Potential. Über den Widerstand fließt zwar ein Strom, der ist aber so klein, daß er nicht weiter beachtet werden soll. Im anderen Fall ist der Taster offen. Die Reset-Leitung liegt über den Widerstand an +5 Volt. Da der fließende Strom nun sehr gering ist, fällt am Widerstand kaum Spannung ab, und die IC-Eingänge liegen praktisch an HIGH-Potential.
Um den geforderten LOW-Pegel im Ruhe-

Bild 1.28: Schaltung des zweistelligen Zählers (0 bis 99) mit zentraler Rückstellung auf 00 (Reset)

zustand zu erreichen, muß der Taster ein Öffner sein, das heißt, er muß in Ruhestellung den Tastkontakt geschlossen halten.

1.8 Darauf können Sie zählen

1.8.1 Der Zähler als Meß- und Kontrollgerät

Eine Zählerschaltung ist beim Experimentieren mit Computern sehr nützlich, weil sich oft das Problem stellt, extrem kurze, in unregelmäßigen Abständen auftretende Impulse registrieren zu müssen. Selbst mit einem Oszilloskop kann man dieser Schwierigkeit nur schlecht beikommen. Daher wollen wir nun ein solches Gerät aufbauen.
Zuvor aber muß noch geklärt werden, woher die Betriebsspannung für unser Bauwerk kommen soll. Man könnte durchaus den Computer als Spannungsquelle heranziehen. Am User-Port stehen beispielsweise stabilisierte 5 Volt zur Verfügung. Sinnvoller ist es jedoch, dem Prüfgerät ein eigenes Netzteil zu spendieren, denn der C64 hat in seiner Versorgung nur wenige Reserven. Messen will man aber oft an Zusatzschaltungen, die den Computer bereits für sich genug belasten. Die Überbeanspruchung des Computernetzteils kann dann zu Störungen führen, die sehr heimtückisch sind und erfahrungsgemäß viel Ärger bereiten.
Auf der Zählerplatine wurde aus diesem Grund gleich ein großzügig dimensioniertes Netzteil eingeplant, das über Steckverbindungen auch andere Geräte speisen kann. Im Verlauf Ihrer späteren Experimente werden Sie erkennen, wie wichtig das ist.

1.8.2 Ein kräftiges stabilisiertes Netzteil

Netzteile werden in Kapitel 7 noch ausführlich besprochen. Bild 1.29, Seite 40, zeigt den Schaltplan der hier verwendeten, stabilisierten TTL-Ausführung. Es besteht nur aus einigen Bauteilen. Das größte davon ist der Transformator. Es wurde eine vergossene Ausführung von Lendermann mit 7,5 Volt Sekundärspannung bei 1,6 Ampere Strom gewählt. Diese Wechselspannung wird gleichgerichtet und über den Elektrolytkondensator C1 geglättet. C2 dient zur Abblockung von Störspannungen. Es folgt ein integrierter Spannungsregler vom Typ 78S05. Obwohl dieser Spannungsregler nur drei Anschlußbeinchen hat, enthält er eine aufwendige Innenschaltung, die nicht nur die vorgesehene Spannung bei jeder Last konstant hält, sondern ihn auch unempfindlich gegen Übertemperatur und Kurzschluß macht. Gekühlt hält dieses Bauelement einen Strom von 2 Ampere aus (das S unterscheidet es von der Normalausführung mit 1A Maximalstrom), es ist jedoch auch ungekühlt durch die eingebauten Schutzschaltungen nicht kaputt zu kriegen – außer bei Verpolen der Anschlüsse.
Auch nach der Regelung folgen noch zwei Kondensatoren, um mögliche Störungen auf der Leitung abzufangen.

1.8.3 Die Netzfrequenz nutzen

Einen Vorteil hat die Netzwechselspannung noch für uns. Sie pulsiert nämlich mit exakt 50Hz. In der Digitaltechnik benötigt man oft Frequenzen, die man sich üblicherweise mit Hilfe von Oszillator-

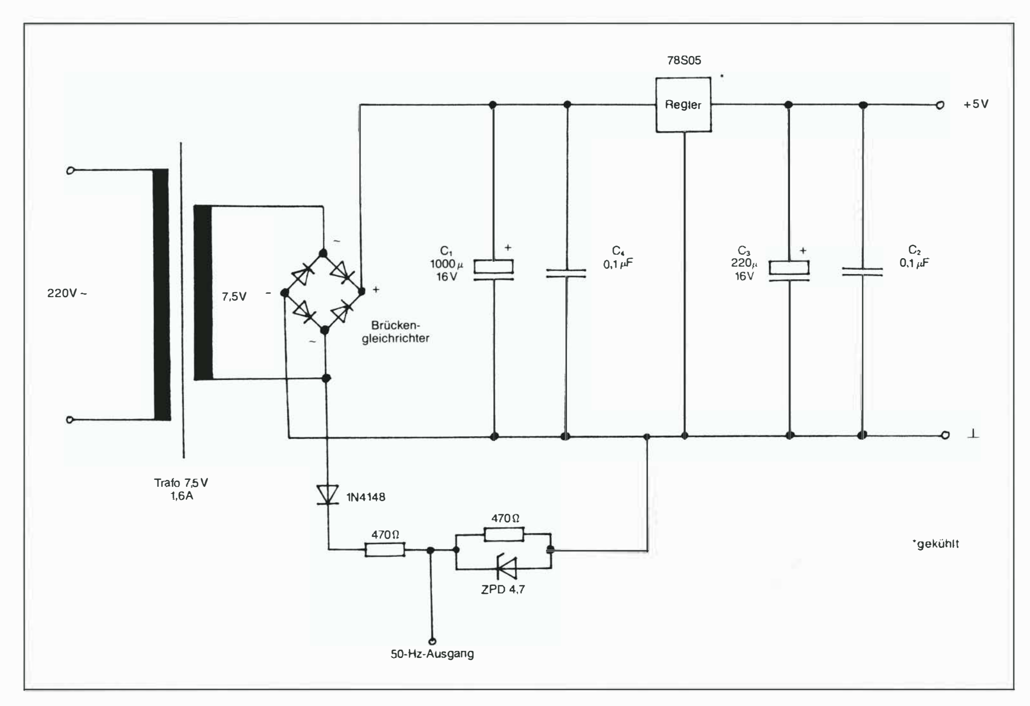


Bild 1.29: Eine Netzteilschaltung für TTL-Bausteine mit Frequenzaufbereitung

schaltungen erzeugt. Daß es auch einfacher geht, zeigt der kleine Zusatz im unteren Teil von Bild 1.29. Über die Diode wird eine sinusförmige Halbwelle aus der herabtransformierten Wechselspannung gewonnen. Die Zenerdiode formt daraus mit Hilfe der beiden Widerstände einen rechteckigen Impuls, der 50mal pro Sekunde auftritt – immer dann nämlich, wenn die Wechselspannung die Zenerspannung von 4,7 Volt übersteigt. Der entstandene einfache 50-Hz-Ausgang kann direkt TTL-Eingänge treiben.

1.8.4 Aufbau einer vielseitigen Test- und Experimentierplatine

Die beschriebenen Einheiten Zähler, Netzteil und Frequenzaufbereiter sind auf der Platine nach Bild 1.30 gemeinsam untergebracht. Man muß jedoch nicht alle Teile bestücken, sondern es steht jedem frei, zum Beispiel das Netzteil wegzulassen. In der Stückliste sind die Bauelemente daher in drei Gruppen eingeteilt. Benötigt man nur den Zähler, dann kann man die Betriebsspannung über zwei Buchsen von außen zuführen. Falls aber der 50-Hz-Ausgang arbeiten soll, ist mindestens noch der Trafo und der Gleichrichter einzusetzen.

Vor den Tips zum Aufbau verweise ich Sie wieder auf Anhang A, der wichtige Informationen über Platinenherstellung, Bauteile und Bestückung enthält.

Bild 1.30 zeigt das Platinenlayout der kombinierten Schaltung. Es besteht aus zwei Teilen. Rechts erkennt man den

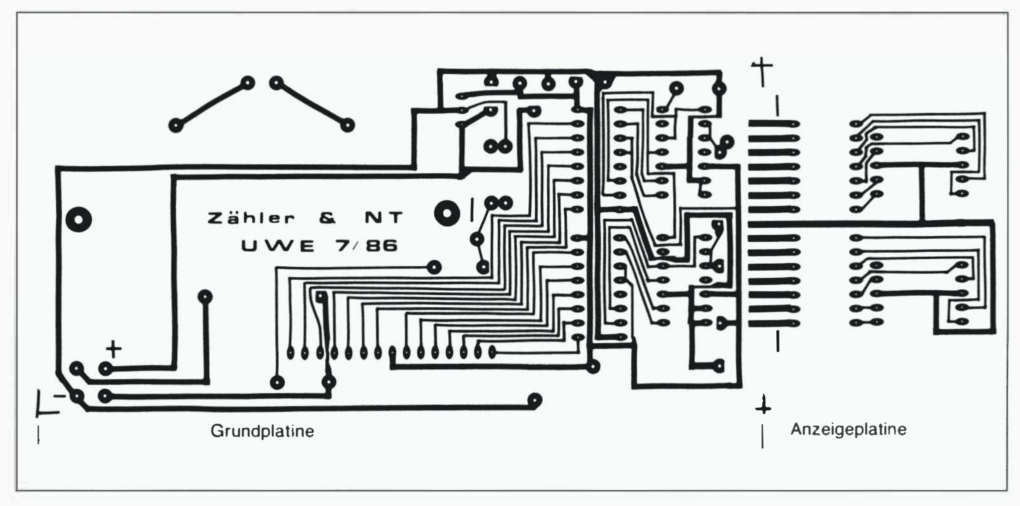


Bild 1.30: Platine für Zähler und Netzteil

Hauptteil und auf der linken Seite die Vorlage für die Anzeigeplatine mit den beiden Sieben-Segment-Anzeigen inklusive ihrer Vorwiderstände. Dieser Teil wird einzeln hergestellt und senkrecht auf die Hauptplatine gelötet, so daß die Anzeige von vorne lesbar ist, wie auf Foto 1.2 zu erkennen. Der ganze Aufbau paßt dann problemlos in das vorgeschlagene Gehäuse mit der durchsichtigen Frontscheibe.

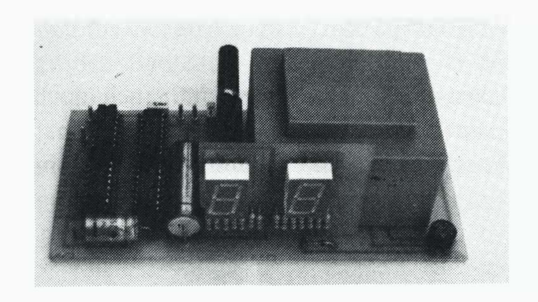


Foto 1.2: Der fertig aufgebaute Zähler

Bitte lesen Sie auch die folgende Aufbauanleitung genau, bevor Sie mit den Arbeiten beginnen und halten Sie die angegebene Reihenfolge exakt ein.

1.8.5 Immer schön der Reihe nach

Vergewissern Sie sich zunächst, ob die Grundplatine in die Aufnahmeschienen des Gehäuses paßt. Sie muß leicht hineinzuschieben und auch wieder herauszuziehen sein. Auch muß die durchsichtige Frontplatte auf das Gehäuse passen. Sollte die Platine zu groß sein, feilen Sie vorsichtig etwas ab.

Bestücken Sie nun die Anzeigeplatine nach dem Plan in Bild 1.31, Seite 42. Um die beiden Platinen später mechanisch genügend fest verbinden zu können, sollten die Anschlußdrähtchen der Vorwiderstände auf der unteren Seite zunächst nur parallel zur Platine umgebogen, aber noch nicht verlötet werden.

Danach widmen Sie sich der Hauptplatine. Vorlage ist wieder der Bestückungsplan (Bild 1.31). Beginnen Sie mit der eingezeichneten Brücke und löten Sie den Trafo noch nicht (!) fest.

Wenn beide Platinen für sich soweit bestückt sind (ohne Trafo!), kann man auf

Eingang
0,1µ
+5V
220V~
Taster
Reset
(Öffner)
74LS90
74LS47
0,1µ
78S05
ZPD
4,7
Brücke
470Ω
1000µ
50 Hz aus
470Ω
Trafo
220V/7,5V
5K6
74LS90
74LS47
220µ
Anzeigeplatine
Gleichrichter
1N4148
Grundplatine
7-
Segment
7-
Segment
14x
330Ω
Anzeigeplatine

Bild 1.31: Bestückung der Zähler-Platinen

die mittlere Kupferbahn zwischen den Widerständen ein zusätzliches Drahtstück anlöten und die 15 überstehenden Enden durch die vorgesehenen Montagelöcher stecken. Foto 1.3 zeigt diesen Schritt. Die Platinen werden so justiert, daß beide Teile rechtwinklig und dicht aufeinanderstehen.

Nun schneiden Sie zunächst die äußeren beiden Enden auf Länge und verlötet sie. Damit hat die Anordnung genügend Stabilität, um nicht mehr so leicht verrutschen zu können. Jetzt sind auch noch kleinere Änderungen der Lage möglich. Anschließend sind die umgebogenen An-

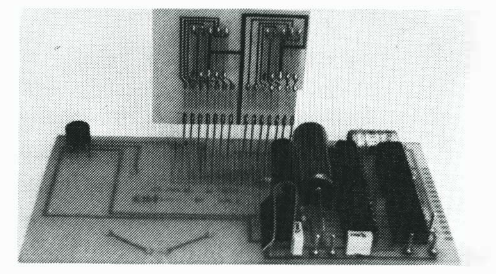


Foto 1.3: Zusammenbau der bestückten Platinen

schlußdrähtchen durch Anlöten auf ihrem gesamten parallel zum Kupfer führenden Weg zu stabilisieren.
Ebenso verfahren Sie jetzt auch mit den übrigen Drahtenden. Achten Sie bitte hier besonders darauf, keine Lötbrücken zu erhalten.
Zum Schluß wird der fehlende Trafo eingebaut, und die Platine ist fertig. Sie können schon einmal ausprobieren, ob auch alles wie auf Foto 1.4 ins Gehäuse paßt.

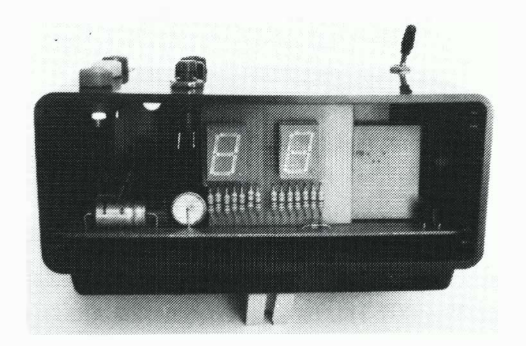


Foto 1.4: Der Zähler nach dem Einbau ins Gehäuse

Zeichnen Sie sich an, wo Löcher für die Buchsen, das Netzkabel und die Bedienelemente gebohrt werden sollen. Achten Sie aber darauf, daß die Buchsen recht weit ins Gehäuse ragen. Dort dürfen keine großen Bauteile – wie zum Beispiel der Trafo – den Weg versperren. Ebenso dürfen die Teile im montierten Zustand auch nicht verhindern, daß die Platine heraus- und hereingeschoben werden kann.
Jeweils zwei Buchsen sind nötig für die 5 Volt Ausgangsspannung und für den Zählereingang (wegen Masse). Beim 50-Hz-Ausgang genügt ein Anschluß. Bedienungselemente sind der Reset-Taster zum Voreinstellen von 00 und eventuell ein Netzschalter.

Nach der Montage dieser Elemente folgt die Verkabelung der Geschichte. Nehmen Sie dazu genügend lange Litzen, um die gesamte Platine auch einmal ganz aus dem Gehäuse ziehen zu können. Peinlich genau müssen Sie darauf achten, daß nirgendwo Netzspannung offenliegt. Schon gar nicht darf die Gefahr bestehen, daß eine an 220 Volt liegende Leitung eine andere bzw. eine der Buchsen berührt! Ziehen Sie also über alle blanken Anschlüsse im 220-Volt-Teil entweder Schrumpfschlauch, oder umwickeln Sie diese Teile sorgfältig mit Isolierband. Auch die beiden geätzten Zuführungen zum Trafo auf der Platine überkleben Sie vorsichtshalber mit Isolierband. Mit Netzspannung ist nicht zu spaßen! Daher in Ihrem eigenen Interesse also den Stecker erst einstecken, nachdem die Platine sicher im Gehäuse ist.
Tabelle 1.8 auf der folgenden Seite enthält eine Aufstellung der benötigten Einzelteile.
Der Zähler läßt sich gut mit dem prellfreien Taster aus Kapitel 1.5 testen. Bei jedem Impuls sollte die Anzeige genau um eine Zahl weiterspringen.
Auch den Frequenzausgang können Sie zum Selbsttest direkt an den Zählereingang anschließen. Es genügt eine Verbindung, da Masse ja bereits intern verbunden ist. Nun muß die Anzeige schnell durchlaufen und nach 99 wieder bei 00 anfangen.
Mit der Reset-Taste kann das Display auf 00 zurückgesetzt werden.
Bisher hatte das Buch anscheinend noch nicht viel mit dem C64 zu tun. Das soll sich jedoch im nächsten Kapitel ändern.

Grundausbau:

1 einseitige Leiterplatte (Grundplatine) nach Bild 1.30

Zähler und Anzeige:

1 Gehäuse ca. 133 x 70 x 56 (Außenmaße) zum Einbau von Uhren oder ähnliches mit durchsichtiger, roter Frontscheibe (z.B. Conrad Electronic Typ 4001)
2 Sieben-Segment-Anzeigen gemeinsame Anode (z.B. HD 1131 A)
2 BCD-Zähler 74LS90
2 IC-Sockel dazu, 14pol
2 Umsetzter 74LS47
2 IC-Sockel dazu, 16pol
14 Widerstände 330 Ohm
1 Widerstand 5,6 Kiloohm
5 Lötnägel
1 Taster (Öffner!)
2 Bananenbuchsen schwarz (für Masse)
1 Bananenbuchse rot (für +5V)
1 Bananenbuchse für Zählereingang (z.B. gelb)
1 einseitige Leiterplatte (Anzeigeteil) nach Bild 1.31

Netzteil:

1 Trafo 7,5 V / 1,6 A (z.B. Landermann Typ 807-1)
1 Spannungsregler 78S05 mit Kühlblech
1 Gleichrichter B80C1500 (runde Bauform)
2 Kondensatoren 0,1 Mikrofarad, Keramik
1 Elektrolytkondensator 1000 Mikrofarad / 16 Volt
1 Elektrolytkondensator 220 Mikrofarad / 16 Volt
1 Knebelschalter für Netzspannung

Frequenzaufbereitung:

1 Zenerdiode 4,7 Volt
1 Diode 1N4148
2 Widerstände 470 Ohm
1 Lötnagel
1 Bananenbuchse für Frequenzausgang (z.B. grün)

Tabelle 1.8: Die Stückliste zur Kombiplatine

Kapitel 2
Der User-Port

Wichtigste Schnittstelle des C 64 und von seiner Anlage her schon für vielfältige Erweiterungen durch den Benutzer gedacht ist der User-Port. Steht der Rechner normal vor uns, dann befindet sich dieser Anschluß links hinten. Bei genauerer Betrachtung ist er eigentlich nichts weiter, als ein Stück Rechnerplatine, an dem 24 Kontakte durch einen Ausschnitt im Gehäuse zugänglich sind. Die Lage und Bedeutung der einzelnen Pins können Sie Anhang C entnehmen.

Schon am überdurchschnittlichen Umfang dieses Kapitels sieht man, daß sich gerade hier unzählige verschiedene Möglichkeiten bieten. Sie werden erfahren, wie man den C 64 auf recht einfache Weise zum Messen, Steuern und Regeln, zur Parametererfassung und Datenausgabe – kurz: zur Kontaktaufnahme mit seiner Umwelt bewegt.

2.1 Geheimdienste im Rechner

Seine Vielseitigkeit verdankt der User-Port zum größten Teil einem Bausteintyp, dem CIA 6526, von dem im C 64 gleich zwei Exemplare eingesetzt wurden. Die Abkürzung CIA hat nichts mit der »Central Intelligence Agency« der vereinigten Staaten zu tun, sondern bedeutet schlicht »Complex Interface Adapter«. Seinem Namen macht dieser Baustein alle Ehre. Er besitzt – wie bei vergleichbaren ICs allgemein üblich – zwei 8 Bit breite Ports zur Ein- und Ausgabe mit Handshake-Möglichkeit. Ebenfalls gängig sind Timer, von denen der 6526 gleich zwei Stück mit je 16 Bit Breite besitzt, die auch noch äußerst universell programmiert werden können. Voraus hat er vergleichbaren Bausteinen aber die Verfügbarkeit einer seriellen Schnittstelle zur wahlweisen Ein- oder Ausgabe von Daten, und quasi als Zugabe besitzt er eine Echtzeituhr mit Alarmzeiterkennung bei einer Auflösung von einer Zehntelsekunde.

2.1.1 Die Verwendung der CIAs im C 64

Wie gesagt, arbeiten im C 64 zwei dieser komplexen Bausteine, deren Möglichkeiten vom System selbst längst nicht voll ausgeschöpft werden. So bleibt glücklicherweise viel Spielraum für eigene Erweiterungen.

Damit Sie sehen, welche Teile der CIAs von eigenen Programmen benutzt werden dürfen und welche nicht, folgt hier bereits kurz eine Aufstellung der Verwendung im

C64. Die Bedeutung der Fachbegriffe wird später noch genauer erläutert.
Der eine CIA 6526 ist im Adreßbereich ab $DC00 (dezimal 56320) ansprechbar (CIA # 1). Er dient im wesentlichen zur Abfrage der Tastatur sowie der Joystick-Ports und zur Erzeugung der Systeminterrupts über den Prozessoranschluß $\overline{\text{IRQ}}$. Das Betriebssystem stellt aufgrund dieser Interrupts übrigens die interne Uhr weiter, die via TI$ von BASIC aus abgefragt werden kann. Sinnvoller – weil wesentlich genauer – wäre die Verwendung einer CIA-Uhr gewesen.
Im einzelnen sieht die Belegung der beiden Ports PA und PB folgendermaßen aus:

Der andere CIA 6526 (CIA # 2) ist ab Adresse $DD00 (dezimal 56576) untergebracht und hat mehrere verschiedene Aufgaben. Er ist für den Datenaustausch über den seriellen Bus verantwortlich, steuert die Lage des Video-RAM im Speicherbereich und bedient den User-Port. Falls der User-Port als RS232-Schnittstelle betrieben wird, erhalten die einzelnen Anschlüsse besondere Funktionen, siehe Tabelle 2.2.
Bei RS232-Betrieb werden die Timer im CIA # 2 zur Baudratengenerierung herangezogen. Damit das Timing exakt stimmt, ist sein Interruptausgang im Unterschied zum CIA # 1 an den nicht maskierbaren Interrupteingang der CPU-$\overline{\text{NMI}}$ gekop-

PA0:	Tastatur Spalte 0 / Joystick-Port 2 Kontakt oben
PA1:	Tastatur Spalte 1 / Joystick-Port 2 Kontakt unten
PA2:	Tastatur Spalte 2 / Joystick-Port 2 Kontakt links
PA3:	Tastatur Spalte 3 / Joystick-Port 2 Kontakt rechts
PA4:	Tastatur Spalte 4 / Joystick-Port 2 Kontakt Feuerknopf
PA5:	Tastatur Spalte 5
PA6:	Tastatur Spalte 6 / Paddle-Set A
PA7:	Tastatur Spalte 7 / Paddle-Set B nur Bit 6 oder 7 =1
PB0:	Tastatur Reihe 0 / Joystick-Port 1 Kontakt oben
PB1:	Tastatur Reihe 1 / Joystick-Port 1 Kontakt unten
PB2:	Tastatur Reihe 2 / Joystick-Port 1 Kontakt links
PB3:	Tastatur Reihe 3 / Joystick-Port 1 Kontakt rechts
PB4:	Tastatur Reihe 4 / Joystick-Port 1 Kontakt Feuerknopf
PB5:	Tastatur Reihe 5
PB6:	Tastatur Reihe 6
PB7:	Tastatur Reihe 7
$\overline{\text{Flag1}}$:	Eingabedaten vom Cassettenport

Tabelle 2.1: Verwendung von CIA #1 $DC00 (56320) im C64

pelt, da dieser die höchste Priorität besitzt. Die CIAs werden daher auch IRQ-CIA (#1) und NMI-CIA (#2) genannt.

2.2 Die I/O-Ports

Beginnen wir nun, den Schnittstellenbaustein CIA 6526 im einzelnen kennenzulernen. Seine diversen Funktionen werden über 16 Adreßregister gesteuert, die in einer Tabelle in Anhang B zusammengestellt sind. Prinzipiell lassen sie sich genauso ansprechen, wie ganz normale Speicherzellen auch, also mit PEEK (Adresse) zum Lesen und POKE Adresse, Wert zum Schreiben. Bei Steuerregistern hat jedoch jedes Bit seine ganz spezielle Bedeutung.

2.2.1 Was dahinter steckt

Port heißt eigentlich Tor oder Hafen und assoziiert in übertragenem Sinn Verbindung oder Anschluß. Wie wir bereits in Tabelle 2.2 sehen konnten, stehen unter anderem an acht Anschlüssen des User-Port die acht Portleitungen PB0 bis PB7 des CIA # 2 zur freien Verfügung. Der andere Port PA ist anderweitig belegt. Die Anschlußzuordnung am User-Port finden Sie in Anhang C. Dabei ist die Ansicht von der Rückseite des Rechnergehäuses her auf den Platinendirektstecker dargestellt.

Relevant für die Zustände dieser acht Portanschlüsse sind die beiden Register 1 und 3 des 6526. Jede einzelne Leitung

PA0: $\overline{\text{VA14}}$ Adreßbit Video-RAM (zweithöchstes)
PA1: $\overline{\text{VA15}}$ Adreßbit Video-RAM (höchstes)
PA2: normal frei, bei RS232-Betrieb TXD (Transmitted Data)
PA3: ATN (Ausgang für seriellen Bus)
PA4: CLOCK (Ausgang für seriellen Bus)
PA5: DATA (Ausgang für seriellen Bus)
PA6: CLOCK (Eingang für seriellen Bus)
PA7: DATA (Eingang für seriellen Bus)

PB0: normal frei, bei RS232-Betrieb RXD (Received Data)
PB1: normal frei, bei RS232-Betrieb RTS (Request To Send)
PB2: normal frei, bei RS232-Betrieb DTR (Data Terminal Ready)
PB3: normal frei, bei RS232-Betrieb RI (Ring Indicator)
PB4: normal frei, bei RS232-Betrieb DCD (Data Carrier Detect)
PB5: frei
PB6: normal frei, bei RS232-Betrieb CTS (Clear To Send)
PB7: normal frei, bei RS232-Betrieb DSR (Data Set Ready)

$\overline{\text{Flag2}}$: normal frei, bei RS232-Betrieb RXD (Receive Data)

Tabelle 2.2: Verwendung von CIA #2 $DD00 (56576) im C64

kann entweder als Eingang oder als Ausgang programmiert werden. Es ist also möglich, die Pins mit logischen Spannungen zu belegen, oder es können die außen anliegenden logischen Werte in den Rechner eingelesen werden. Der Baustein reagiert dabei genau wie die in Kapitel 1 vorgestellten TTL-Gatter. Spannungen nahe 0 Volt ergeben logisch 0 oder LOW, Spannungen nahe 5 Volt logisch 1 bzw. HIGH.

2.2.1.1 Das Datenrichtungsregister

Die Festlegung, welche Leitung Eingang und welche Ausgang sein soll, geschieht über das Datenrichtungsregister. Für den User-Port (PB) maßgeblich ist das Register Nummer 3. In ihm steht jedes Bit für eine bestimmte Leitung. Bit 0 steuert PB0, Bit 1 PB1 und so weiter. Ist ein Bit gesetzt (1), so wird die entsprechende Leitung zum Ausgang, ein gelöschtes Bit programmiert sie auf Eingang.
Die Basisadresse des CIA # 2, der auch den User-Port bedient, ist $DD00, also dezimal 56576. Um Register 3 anzusprechen, addieren wir einfach drei zu dieser Adresse. Ein Befehl, der alle acht Portleitungen auf Ausgabe setzt, lautet also

```
POKE 56576+3,255 ,
```

denn 255 ist die größte mit 8 Bit darstellbare Zahl, bei der somit alle Bits den Wert 1 haben.

2.2.1.2 Das Datenregister

Ganz analog sind die Bits des Datenregisters den einzelnen Leitungen zugeordnet. Das Datenregister hat die Nummer 1. Wir erreichen es also durch Addition von 1 zur Basisadresse von CIA # 2. Jedes gesetzte Bit erzeugt hier auf der entsprechenden Ausgabeleitung eine Spannung mit HIGH-Pegel (nahe +5 Volt), jedes gelöschte Bit jedoch LOW-Pegel (nahe 0 Volt).
Wollen wir beispielsweise die Leitung PB0 auf HIGH-Pegel setzen, alle anderen Leitungen sollen aber auf LOW gehen, dann lautet die Anweisung dazu:

```
POKE 56576+1,1 ,
```

denn die Dezimalzahl 1 wird durch die Dualkombination

```
0 0 0 0 0 0 0 1
```

dargestellt (siehe Kapitel 1 und Anhang E). Die 0 ganz links steht dabei für PB7, man sagt auch das höchstwertige Bit (englisch Most Significant Bit: MSB), die ganz rechts für PB0, das niederwertigste Bit (Least Significant Bit: LSB).
Natürlich müssen dazu alle Leitungen als Ausgänge definiert sein. Macht man sie durch Löschen aller Datenrichtungsbits mit

```
POKE 56576+3,0
```

ausnahmslos zu Eingängen, dann kann man die jeweils außen am User-Port anliegenden aktuellen logischen Zustände per Programm abfragen, indem man mit

```
PEEK (56576+1)
```

das Datenregister einliest.
Ein kleines BASIC-Programm zum Abfragen der Portleitungen ist also schnell geschrieben und lautet zum Beispiel:

```
10 POKE 56576+3,0:
REM ALLE PORTLEITUNGEN AUF EINGABE
20 PRINT PEEK (56576+1):
REM WERT AUSGEBEN
```

Listing 2.1 Minimalprogramm zur Abfrage des User-Ports

Wenn Sie das Programm starten, ohne irgendetwas am User-Port angeschlossen zu haben, erhalten Sie als Antwort 255. Daraus folgt, daß alle Portanschlüsse auf HIGH-Pegel liegen. Man erkennt, daß sich der 6526 zumindest in diesem Punkt tatsächlich wie ein TTL-Baustein verhält. Wie wir aus Kapitel 1 bereits wissen, waren auch dort offene Eingänge mit HIGH-Potential gleichzusetzen.
Da aber offene Eingänge allein schnell uninteressant werden, sollen nun einige Beschaltungsmöglichkeiten erläutert werden.

2.2.2 Sicherheitsmaßnahmen beim Experimentieren mit dem Port

Bevor Sie sich mit dem Lötkolben bewaffnen und gnadenlos den Rechner traktieren, zunächst noch ein Wort zum CIA. Wie Sie gesehen haben, ist dieser Baustein in der Tat komplex, was für uns als Bastler den Nachteil mit sich bringt, daß er aufgrund der hohen Integrationsdichte auch relativ empfindlich ist. Das gilt nicht nur für statische Elektrizität (genaueres steht in Anhang A 4.3.4), sondern in verstärktem Maße auch für die Einhaltung der maximal zulässigen Ausgangsströme. Direkt kann er pro Ausgangspin nur zwei TTL-Lasten treiben. Sollte der Strom irgendwann zu groß werden, wird der Baustein ohne Vorwarnung zerstört. Zum Glück ist der 6526 inzwischen schon ohne große Schwierigkeiten in vielen Geschäften zu haben, doch muß man für ihn meist immer noch knapp DM 40,- hinblättern.
Dieser Ärger läßt sich bei sorgfältigem Vorgehen mit Sicherheit vermeiden. Eine erste Richtlinie dazu lautet folgendermaßen:
Voraussetzung für alle Experimente ist ein User-Port-Stecker, an den Kabel oder Platinen angelötet werden können. Dieser Stecker darf nie auf den User-Port aufgesteckt oder abgezogen werden, während der C64 eingeschaltet ist, da hierbei Kurzschlüsse entstehen könnten.
Die zweite Regel kommt bei Schaltungen zum Tragen, die selbst mit ihrem Ausgang an den Port angeschlossen werden. Wie schon bei TTL-Bausteinen, muß genau darauf geachtet werden, daß nicht zwei Ausgänge gegeneinander arbeiten. Über kurz oder lang würde mindestens einer der Ausgangstransistoren zerstört werden. Man sollte also sicherheitshalber auch unbenutzte Portanschlüsse immer als Eingang programmieren. Aus diesem Grund wird auch beim Einschalten des Rechners (und bei jedem anderen Reset) das gesamte Datenrichtungsregister gelöscht. Alle Portleitungen befinden sich so mit Sicherheit im Eingangsmodus, bis das Betriebssystem oder der Anwender sie je nach ihrer Bestimmung programmiert. Diesen Vorgang nennt man Initialisierung.
Eine weitere Regel betrifft die angeschlossene Schaltung selbst:
Will man externe Geräte oder hohe Spannungen vom Computer aus schalten, ist es unbedingt ratsam, den Weg über ein Relais oder einen Optokoppler zu wählen. So schützt man seine wertvolle Ausrüstung am wirkungsvollsten vor möglichen Schä-

den. Beide Schaltungsprinzipien trennen die Last galvanisch vom Portausgang, das heißt, es existiert keine leitende Verbindung zwischen Computer und zu schaltendem Gerät. Damit braucht man nicht ständig zu befürchten, daß der maximal zulässige Schaltstrom überschritten und der Portbaustein zerstört wird, oder daß über ein angeschlossenes 220-Volt-Gerät eventuell gar Netzspannung am System anliegt. Den Lastteil sollte man – sofern er irgendwie mit Netzspannung zu tun hat – ohnehin in ein gut isoliertes Gehäuse einbauen, auch wenn er »nur zum Ausprobieren« aufgebaut wurde. Jegliche Berührung mit Netzspannung führenden Teilen ist lebensgefährlich! Dessen sollte man sich ständig bewußt sein.

2.2.3 Datenausgabemöglichkeiten

Im folgenden Abschnitt werden einige Grundschaltungen angegeben, über die in irgendeiner Weise die Umwelt des Computers beeinflußt werden kann. Sie kommen teilweise bei späteren Platinen zum Einsatz.

2.2.3.1 Der Computer schaltet per Programm!

Wer ohne viel Aufwand testen will, ob das bisher über den Port Gesagte auch tatsächlich stimmt, kann einfach mit einem Vielfachmeßgerät die Spannungen an den Portanschlüssen nachmessen. Die meist schwarze Strippe für den Minuspol klemmt man dazu an einen der vier Masseanschlüsse GND (Ground), von denen sich je einer am äußersten Kontakt beider Anschlußreihen befindet. Die andere Strippe kommt an einen beliebigen Portanschluß PB0 bis PB7 (Pin C bis L an der unteren Kontaktleiste).
Schaltet man nun wie oben erläutert mit

```
POKE 56576+3,255
```

alle Leitungen auf Ausgabe und legt dann zunächst mit

```
POKE 56576+1,255
```

diese Ausgänge alle auf HIGH, dann kann man an dem ausgewählten Meßpunkt eine Spannung knapp unter +5 Volt messen.
Nun bringt

```
POKE 56576+1,0
```

alle Portausgänge wieder auf LOW. Die Nadel des Meßgeräts schwingt zurück und zeigt jetzt einen kaum meßbaren Wert dicht bei 0 Volt.
Probieren Sie die letzten beiden POKE-Befehle mehrmals hintereinander aus. Der Zeiger muß jedesmal hin- und herschwingen. Dasselbe geschieht natürlich auch an allen anderen Portleitungen.
Wenn Sie schon den Zähler aus Kapitel 1.8 aufgebaut haben, dann schließen Sie seinen Eingang nun an einen der Ausgänge an. Natürlich muß auch die Masse-Buchse mit der Computermasse verbunden werden (wie das Meßgerät).
Sie sehen deutlich, daß die Anzeige bei jeder HIGH-LOW-Flanke weiterspringt, im umgekehrten Fall jedoch unverändert bleibt.

2.2.3.2 Pegel sichtbar – Datenausgabe über LED

Eine Leuchtdiodenanzeige der anliegenden Spannung ist sehr leicht zu realisieren.

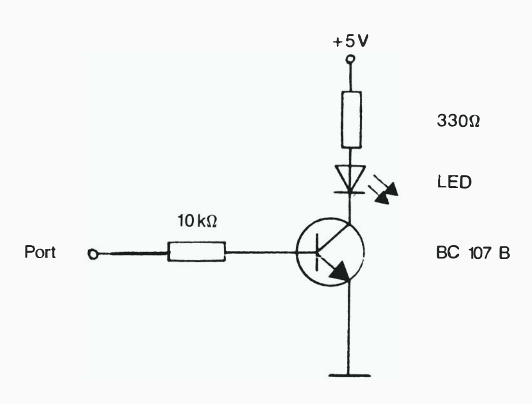


Bild 2.1: Ansteuerung einer Leuchtdiode

Allerdings sollte man sicherheitshalber einen Transistor vorschalten, um den Strom zu begrenzen. Bild 2.1 zeigt die ganze Schaltung. Sie funktioniert folgendermaßen:
Die Leuchtdiode (Light Emitting Diode = LED) ist über einen Vorwiderstand und den Transistor zwischen +5 Volt und Masse geschaltet. Leuchtdioden müssen immer über einen Vorwiderstand betrieben werden, der den Strom begrenzt und damit gleichzeitig die Helligkeit einstellt.
Der Transistor wird hier als ein vom Basisanschluß steuerbarer Schalter betrieben und liegt ebenfalls über einen Basisvorwiderstand am Port. Führt der Portausgang HIGH-Spannung, so nimmt die Kollektor-Emitter-Strecke des Transistors einen sehr kleinen Widerstand an. Denkt man sich den Transistor als Schalter, so ist dieser geschlossen. Es fließt ein genügend großer Strom, um die LED zum Leuchten zu bringen.
Sinkt die Portspannung auf LOW, dann steigt der Widerstand des Transistors und es kann kaum noch Strom durch ihn fließen. Der gedachte Schalter ist jetzt geschlossen. Also erlischt die Leuchtdiode.

Schauen Sie sich noch einmal Bild 1.11 an. Es zeigt die Innenschaltung eines TTL-Gatters mit offenem Kollektor. Auch hier wird der Ausgang über einen Schalttransistor gesteuert. Daher kann man statt diskreter Elektronik – also statt einem Transistor mit Vorwiderstand – auch entsprechende ICs verwenden.

2.2.3.3 Krach macht's

Nicht nur Leuchtdioden können von einem Schalttransistor gesteuert werden, sondern schlicht alles, was Strom braucht. Dazu gehört auch ein Piezo-Summer mit eingebautem Tongenerator. Er wird ohne Vorwiderstand betrieben. Sobald der Transistor durchschaltet, das heißt, wenn HIGH an seiner Basis liegt, macht der Summer Krach. Natürlich kann auch der Sound-Chip des Computers bei geeigneter Programmierung summen, viel schöner sogar, diese Funktion ist aber beispielsweise immer dann sinnvoll, wenn bei langen Programmlaufzeiten der Monitor abgeschaltet oder der Fernseher anderweitig benutzt werden soll. Durch einfaches Einschalten eines Portbit meldet sich der Computer, auch wenn ihm kein Verstärker zur Verfügung steht.
Bild 2.2 zeigt diese Schaltung.

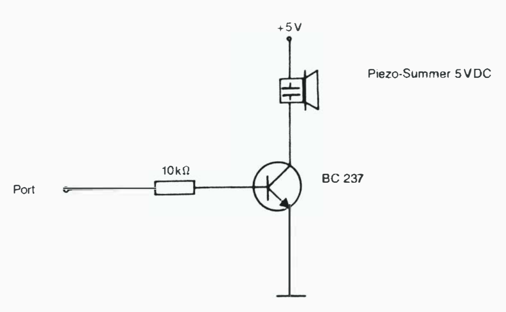


Bild 2.2: Ansteuerung eines Summers mit eingebautem Tongenerator

2.2.3.4 Da geht's rund

Sie ahnen es schon. Natürlich kann man einen Elektromotor in den Kollektorkreis setzen, und er beginnt sich zu drehen, wenn der Transistor den Strom einschaltet. Doch was passiert beim Abschalten? Der Motor wird nicht sofort stehenbleiben, sondern noch einige Umdrehungen machen. Nun ist aber ein Motor grundsätzlich nichts anderes als ein Generator. Wenn sich die Achse mit den Wicklungen im Magnetfeld dreht, entsteht ein Strom, der nun durch den Transistor abfließt und durchaus in der Lage ist, den Transistor zu zerstören. Um solche Ströme abzuleiten, wird üblicherweise eine Diode eingefügt. Bild 2.3 zeigt, wie man einen Motor komfortabel steuern kann. Damit die Logikschaltung nicht belastet wird, und damit ein Betrieb mit höheren Spannungen möglich ist, wird eine getrennte positive Spannung am Punkt +UM eingespeist. Sie wird dem Motor über jeweils zwei Transistoren zugeführt, die mittels zweier TTL-Gatter mit offenem Kollektor gesteuert werden. Nur so ist auch der Betrieb mit höheren Spannungen möglich. Durch die Art der Verschaltung kann der Motor vorwärts und rückwärts laufen, wenn ein Gatter HIGH, das andere LOW ist. Neben den uns schon bekannten NPN-Transistoren wurden nämlich auch die komplementären PNP-Typen eingesetzt, wodurch sich eine einfache Ansteuerung ergibt. Ein NPN-Transistor (oben) schaltet immer dann durch, wenn an seiner Basis HIGH-Pegel liegt, ein PNP-Transistor (unten) dagegen nur bei LOW-Pegel. Der Motor hält an, wenn beide Gatter gleich angesteuert

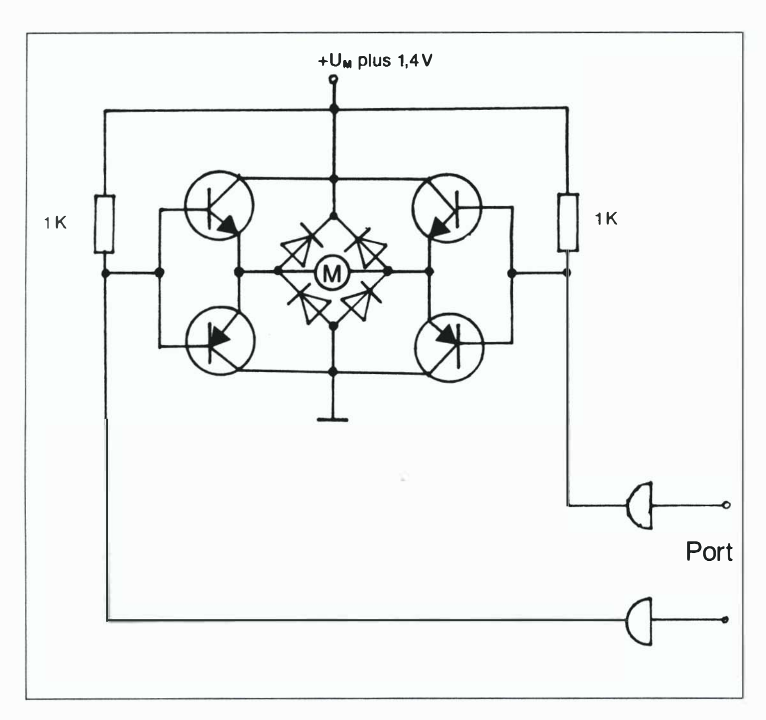


Bild 2.3: Motorsteuerung mit dem Computer

werden. In diesem Fall befinden sich beide Pole des Motors auf gleichem Potential. Die Transistortypen hängen von der Größe des angeschlossenen Motors ab. Sie müssen die dabei entstehenden Ströme sicher schalten können.

Diese interessante Schaltung befindet sich unter anderem auf der MultiI/O-Platine in Kapitel 2.2.6. Dort finden Sie auch nähere Angaben.

2.2.3.5 Ansteuerung von Schrittmotoren

Ein Schrittmotor wandelt elektrische Impulse in mechanische Drehbewegung mit definiertem Drehwinkel um. Er stellt ein mechanisches Bauelement dar, dessen Achse, den Steuerimpulsen folgend, schrittweise rotiert. Schrittmotoren werden überall dort eingesetzt, wo genau festgelegte und reproduzierbare Bewegungen nötig sind, beispielsweise als Antrieb für den Schreib-/Lesekopf in Floppylaufwerken oder des Druckkopfes in Druckern, zur Blendensteuerung in Kameras, in Kurven-Lochstreifen- oder Kartenschreibern sowie in Robotern.

Jedesmal, wenn der Schrittmotor geeignet angesteuert wird, dreht sich seine Achse um einen durch die Bauart genau festgelegten Winkel. Übliche Schrittmotoren mit kleiner Schrittzahl drehen die Achse bei jedem Schritt um 7,5, 15, 45 oder 90 Grad. Motoren mit höherer Schrittzahl haben Standardschrittweiten von 1,8 und 5,0 Grad. Bei bekanntem Anfangspunkt der Bewegung läßt sich die Position der Achse zu jeder Zeit genau angeben. Um eine bekannte Anfangsposition zu erreichen, wird üblicherweise die Welle an einen Anschlag gefahren. Alle nachfolgenden Impulse haben nun keine Wirkung mehr, und man kann nach diesem Vorgang sicher von der vorgesehenen Position ausgehen. Drucker und Floppylaufwerke kalibrieren meist auf diese Weise ihre Kopfposition.

Der Schrittmotor selbst besitzt in der Regel vier Spulen, bezeichnet mit S_1, S_2, S_3 und S_4. Werden sie in geeigneter Weise mit Stromimpulsen angesteuert, dann führt der Motor einen Schritt aus. Dazu gibt es drei mögliche Verfahren:

- Wenig Leistungsbedarf
- Normal
- Halbschrittverfahren

Bei der ersten Möglichkeit wird ein Stromimpuls der Reihe nach durch alle vier Spulen geschickt: zuerst durch S_1, dann durch S_2, durch S_3, durch S_4 und dann wieder durch S_1 und so fort. Bild 2.4 macht das deutlich. Nie werden zwei Spu-

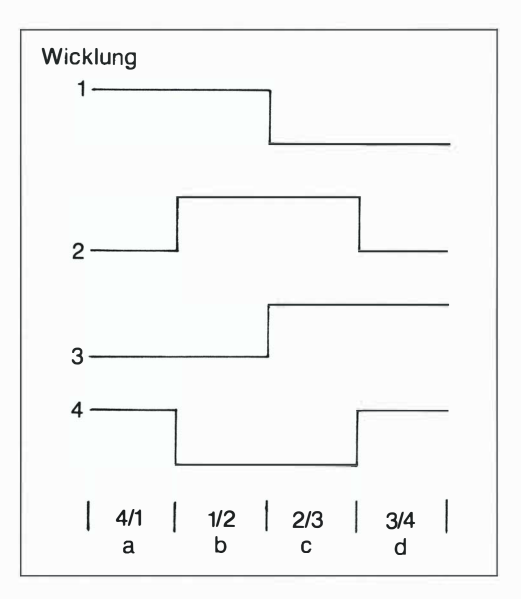


Bild 2.4: Zeitablauf der Wicklungsansteuerung im Normalverfahren

len zur selben Zeit durchflossen und der Motor macht einen Schritt pro Impuls.

Das Normalverfahren steuert immer zwei Wicklungen zugleich in folgender Reihenfolge an: S_1 und S_2, S_2 und S_3, S_3 und S_4, S_4 und S_1 und so weiter. Bild 2.4 macht das deutlich. Es ergibt sich eine sanftere Arbeitsweise des Motors, jedoch wird mehr Leistung benötigt.

Das Halbschrittverfahren erlaubt, zwischen jeden ganzen Schritt einen halben einzufügen. Die Ansteuer-Reihenfolge lautet hier: 1 und 2, 2, 2 und 3, 3, 3 und 4, 4, 4 und 1, 1 und so fort.

Die Spulen des Motors benötigen eine besondere Stromquelle, da sie mit typisch 0,2 Ohm einen recht kleinen Innenwiderstand besitzen. Wegen der andererseits hohen Induktivität der Spulen sind besondere Entwurfstechniken notwendig, um die Schalttransistoren und Filterkondensatoren vor Zerstörung durch die induzierten Spannungsspitzen zu schützen. Bild 2.5 zeigt eine Interfaceschaltung für Schrittmotorbetrieb am User-Port.

Mit dem IC SAA 1027 von Siemens kann der Ansteuervorgang wesentlich erleichtert werden. Bild 2.6 zeigt die zugehörige Schaltung. Es ist nur noch ein Impulseingang nötig (Pin 15), über den mittels einer Frequenz die Drehgeschwindigkeit des

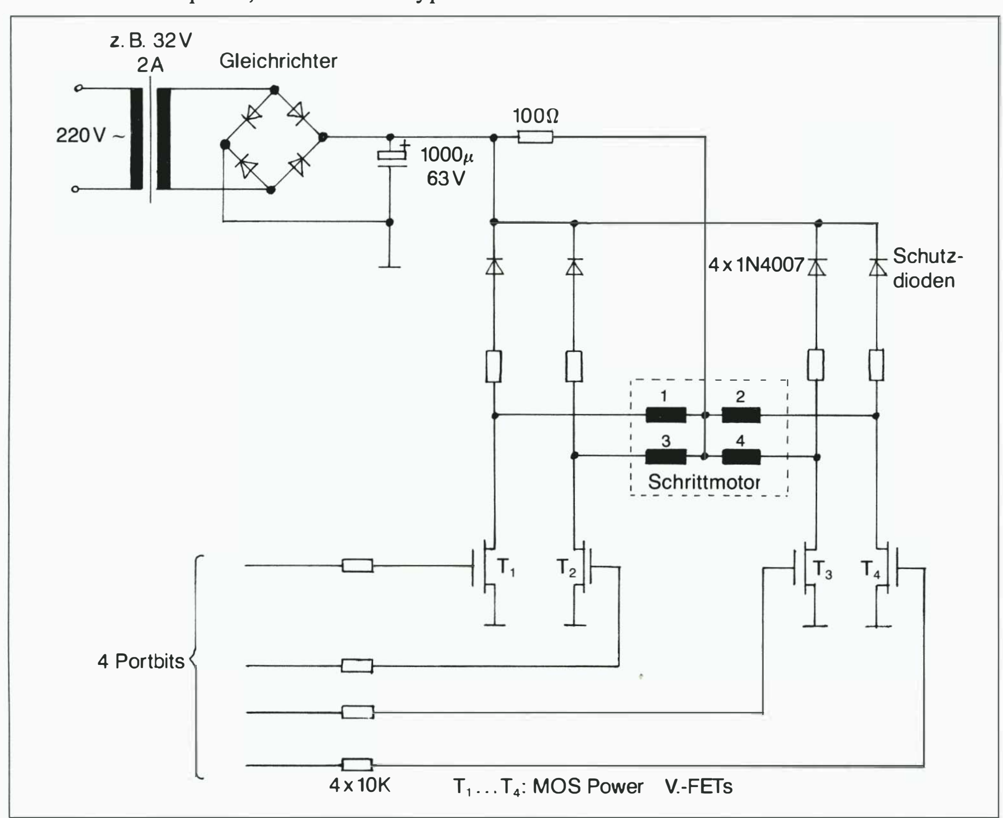


Bild 2.5: Ansteuerung eines Schrittmotors

Bild 2.6: Schrittmotoransteuerung mit dem IC SAA 1027

Motors bestimmt wird. An Pin 3 legt HIGH- oder LOW-Pegel die Drehrichtung fest.

2.2.3.6 Galvanisch getrennt, doch funktionell vereint

Ebenso wie in der LED- und der Summerschaltung, liegen die Verhältnisse auch bei der Relaisschaltung nach Bild 2.7. Der Transistor schaltet wieder den Strom. Dabei darf keinesfalls die Schutzdiode 1N4148 parallel zum Relais vergessen werden. Sie schließt negative Spannungsspitzen kurz, die – ähnlich wie beim Motor – durch Abfallen des Ankers im Inneren der Relaisspule aufgrund von Induktionswirkung entstehen und den Schalttransistor zerstören könnten.

Durch die Spule des Relais erfolgt bei dieser Anordnung die Kopplung von Ursache

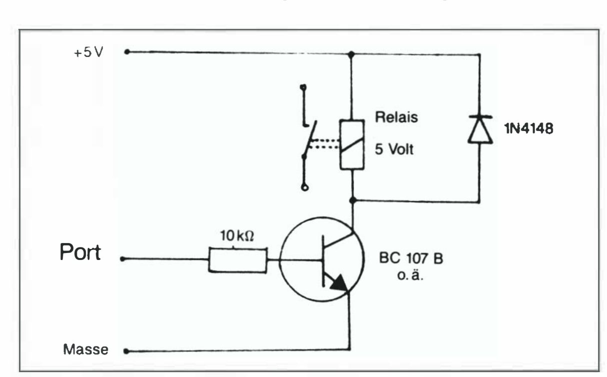


Bild 2.7: Schalten mit Relais am Computer

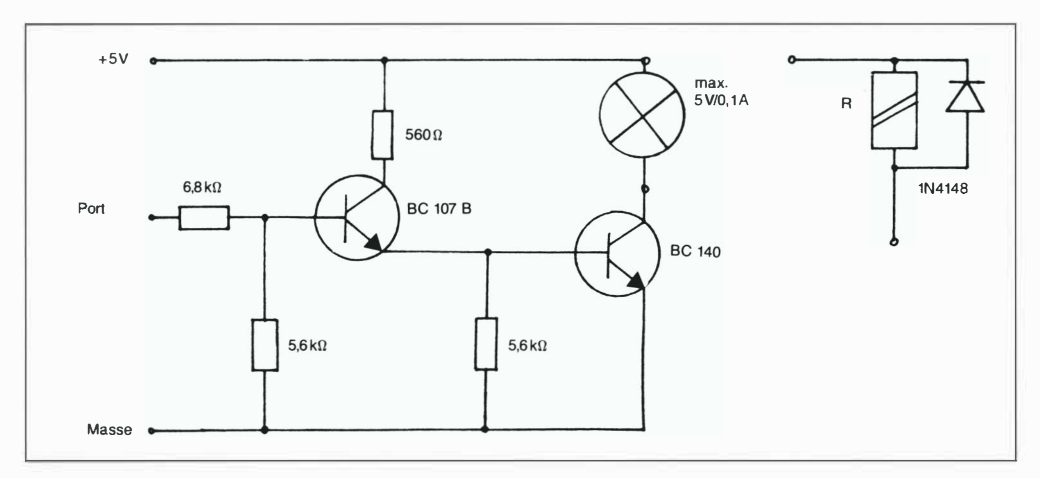


Bild 2.8: Galvanisch gekoppelte (!) Schaltung für größere Lasten, zum Beispiel starke Relais (anstelle der Glühbirne)

und Wirkung nicht galvanisch, das heißt elektrisch leitend, sondern über ein magnetisches Feld. Sobald Strom durch die Wicklung fließt, zieht der Anker an, und der Lastkontakt wird geschlossen.

Für größere Lastströme, wie sie zum Beispiel in Glühlämpchen oder stärkeren Relais für höhere Schaltleistungen leicht auftreten können, muß die Anordnung nach Bild 2.8 aufgebaut werden. Auch hier – wie immer beim Einsatz von Spulen und bewegten Teilen – die Schutzdiode bitte nicht vergessen!

Über die Relaiskontakte ist es nun möglich, die unterschiedlichsten Geräte zu schalten. Dabei ist der Lastkreis galvanisch völlig unabhängig vom Steuerkreis (Computer), so daß ohne weiteres auch Wechselströme schaltbar sind.

2.2.3.7 Die Sache mit dem Optokoppler

Eine andere Realisierungsmöglichkeit der galvanischen Trennung ist die Kopplung mittels Licht. Bild 2.9 zeigt strenggenommen zwei Schaltungen. Links ein gewöhnlicher Anzeigekreis mit einer Leuchtdiode nach Bild 2.1. Die LED leuchtet bei HIGH-Pegel und erlischt bei LOW am Steueranschluß. Dicht neben dieser Leuchtdiode befindet sich ein Fototransistor, der über einen Thyristor zum Beispiel Netzspannung schaltet. Die Gleichrichterbrücke ermöglicht eine Ausnutzung aller Halbwellen des Wechselstroms.

Solche und ähnliche kontaktlose, gleichspannungsgesteuerte Wechselstromschaltstufen finden bei Lichtsteuerungen, Heizungsregelungen oder Motorsteuerungen häufig Verwendung.

Die Anordnung Leuchtdiode-Fototransistor gibt es auch fertig als Optokoppler, lichtdicht eingegossen in ein gemeinsames Gehäuse. Seine Anschlußbelegung – wie auch die des verwendeten Thyristors – finden Sie im Anhang A.

Ein Thyristor hat die Eigenart, immer nur eine Halbwelle durchzulassen. Daher werden Doppelthyristoren – sogenannte Triacs – angeboten, die diesen Nachteil

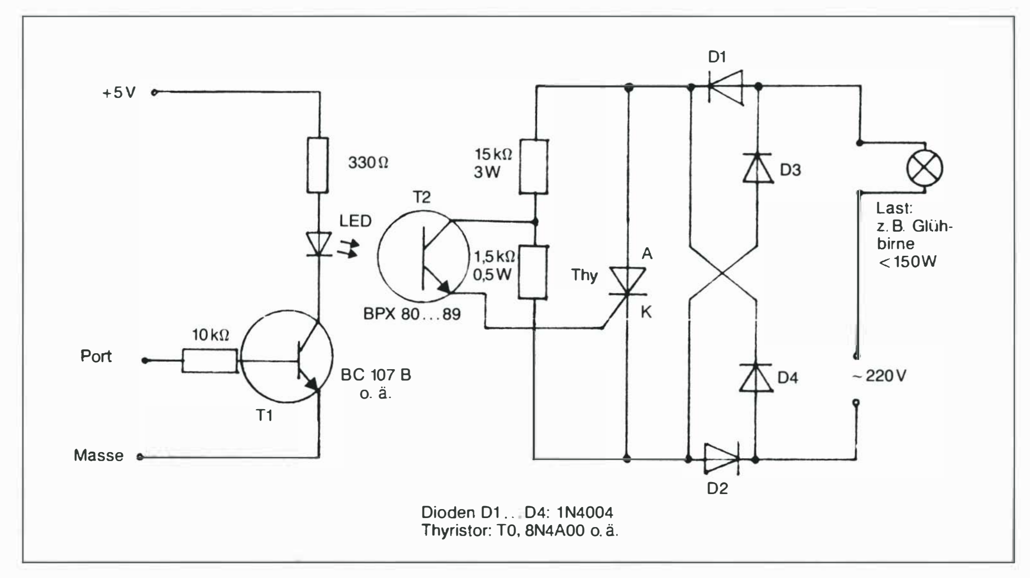


Bild 2.9: Schalten einer 220-Volt-Last, galvanisch vom Computer getrennt, ohne Relais (optische Kopplung)

beheben. Bild 2.10 zeigt eine entsprechende Schaltung, durch die beide Halbwellen ausgenutzt werden. Zusätzlich sorgen einige passive Bauelemente für eine wirkungsvolle Entstörung.

Auch hier erfolgt die Ansteuerung mittels Licht.

Ganz besonders einfach kann man sich's machen, wenn die zu schaltende Leistung nicht größer als 66 Watt ist. In diesem

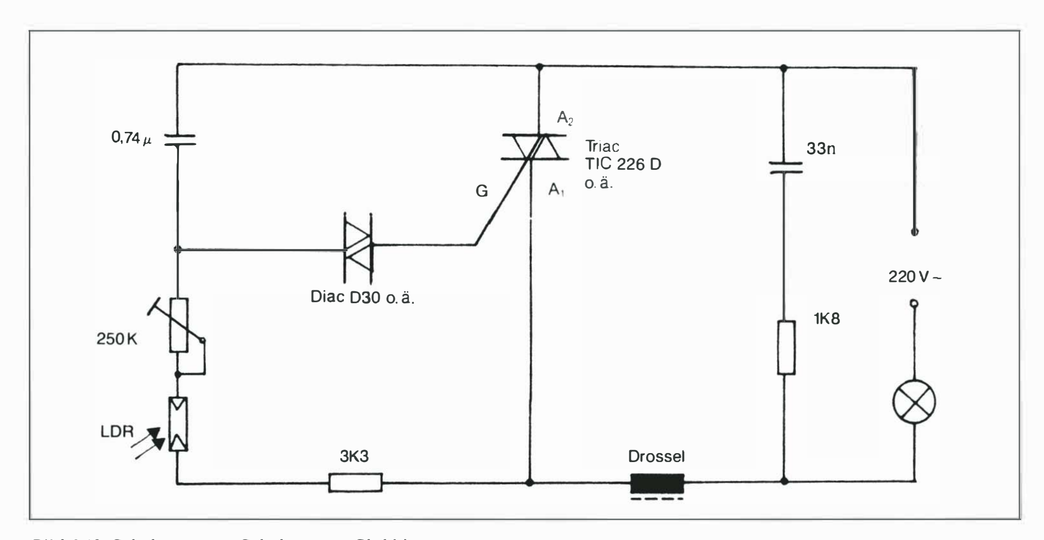


Bild 2.10: Schaltung zum Schalten von Glühbirnen

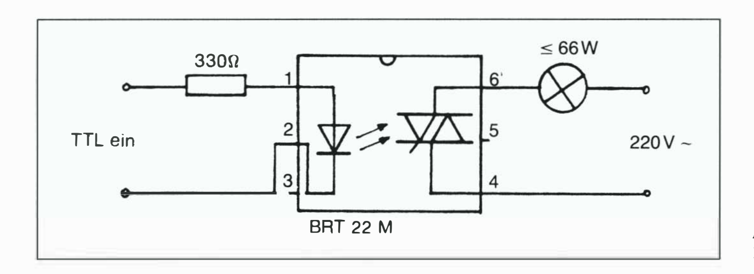


Bild 2.11: Besonders einfache 220-V-Lastschaltung

genügt der Leistungsschalter BRT22M. Bei äußerst kompakter Bauart enthält dieses integrierte Bauelement Optokoppler, Lastteil und Entstörung in einem Gehäuse. Bild 2.11 zeigt bereits die gesamte Schaltung.

2.2.3.8 Warum nicht auch mal regeln?

Die zuletzt beschriebenen Anordnungen bieten gegenüber den Relaisschaltungen neben der entfallenden mechanischen Kontaktbelastung vor allem den Vorteil größerer Schaltgeschwindigkeit. »Was hat man aber davon?« könnten Sie nun fragen, »wenn man etwas so schnell schalten kann, daß man gar nicht mehr weiß, ob es gerade ein oder aus ist?«

Doch damit ergeben sich vollkommen neue Möglichkeiten:

Schaltet man den Ansteueranschluß mit dem Computer sehr schnell ein und aus, kann man über die Schaltfrequenz eine angeschlossene Glühbirne beispielsweise rechnergesteuert dimmen, das heißt praktisch in ihrer Helligkeit regeln.

Das Zeitdiagramm in Bild 2.12 veranschaulicht diese Betriebsart für die Thyristor-Schaltung:

Sobald die Steuerspannung am Port HIGH-Pegel erreicht, leuchtet die LED auf, und der Thyristor schaltet sofort durch: Im Lastkreis kann Strom fließen. Anders beim Umschalten der Portspannung auf LOW. Die Leuchtdiode erlischt zwar sofort, eine Eigenschaft des Thyristors ist es aber, erst ab dem nächsten Nulldurchgang der Netzspannung wieder zu sperren. Die Last wird in Wirklichkeit mit Wellenpaketen betrieben.

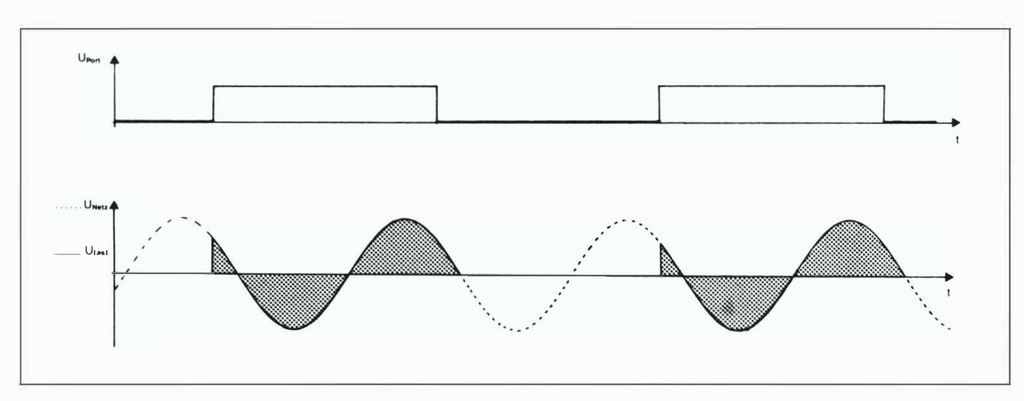


Bild 2.12: Zeitdiagramm zum Schaltverhalten der Anordnung aus Bild 2.9

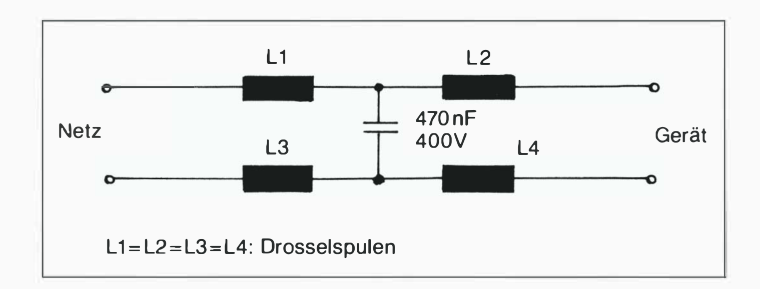


Bild 2.13: Entstörglied gegen Schaltspitzen

Falls die Schaltpausen kurz genug gewählt sind, merkt man das bei den meisten Verbrauchern nach außen hin nicht, da zum Beispiel der Glühfaden in einer Lampe immer noch gewisse Zeit nachleuchtet und damit die zu erwartende Flackerwirkung kompensiert.

Denkbar wäre als Steuerprogramm eine Interruptroutine, die bei jedem Aufruf den Portanschluß triggert, also zunächst einschaltet und gleich wieder ausschaltet. Der Lastkreis bleibt dann bis zum nächsten Nulldurchgang eingeschaltet. Wenn ein Timer diesen Interrupt auslöst, kann über den Zählerinhalt die Helligkeit gesteuert werden, während der Rechner wie gewöhnlich für andere Aufgaben zur Verfügung steht. Schließt man an mehrere Kanäle verschiedenfarbige Lampen an, können herrliche Lichtspielereien mit weichen Übergängen erzielt werden.

Ein lohnendes Einsatzgebiet für Fotofreunde wäre die Ansteuerung zweier (oder mehrerer) Diaprojektoren, die gemeinsam eine vom Rechner gesteuerte Show mit allen Schikanen absolvieren könnten. Dazu müßte je ein Kanal in der beschriebenen Weise für die Dimmerung der Projektorlampen ausgebaut werden, ein anderer mit der Relaisschaltung nach Bild 2.7 für die Steuerung des Transports.

2.2.3.9 Was tun, wenn's stört?

Gleich noch ein Tip: Sollten Sie Probleme mit dem »Einschaltknacks« der gesteuerten Geräte haben, wenn also Ihr Computer beim Schalten von großen Lasten manchmal »aussteigt«, dann hilft folgende Entstörmaßnahme:

Ein Netzfilter nach Bild 2.13 wird in die Zuleitung des angeschlossenen Geräts gebracht. Jetzt können kurzzeitige Spannungseinbrüche über dessen Versorgungsleitung nicht mehr zum Netz gelangen und den Computer aus dem Tritt bringen. Als Drosselspulen L1...L4 eignen sich beispielsweise die Typen SFT 1030. Genau diese Schaltung hat sich auch bewährt, wenn der Computer jedesmal beim Einschalten des Kühlschranks den Hut nimmt. Auch fertige Netzfilter – ausgebildet etwa als Kaltgerätestecker – werden angeboten. Leider sind sie meist nicht gerade billig. Bild 2.14 zeigt die Innenschaltung eines solchen Filters. Seine Funktionsweise beruht auf der Tatsache, daß der Innenwiderstand einer Spule mit steigender Frequenz zunimmt. Dagegen stellt ein Kondensator für hochfrequente Störimpulse praktisch einen Kurzschluß dar.

Eine Unterdrückung von Störungen im Netz durch impulsweise Trafo- Belastung kann eventuell bereits durch einen kleinen

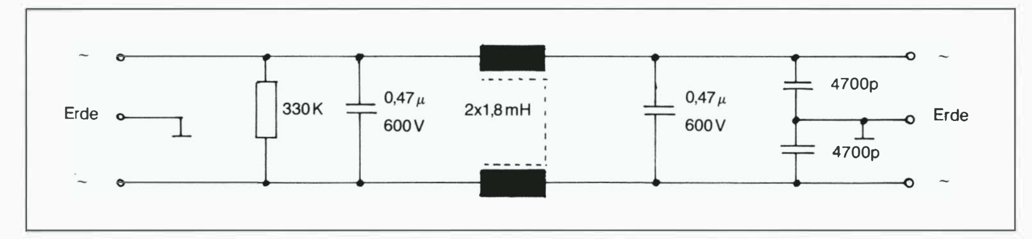


Bild 2.14: Schaltung eines Netzfilters

Widerstand von etwa 0,5 bis 1 Ohm behoben werden, der in Serie zur Sekundärwicklung des Trafos geschaltet wird.
Als einfachste Möglichkeit sollten Sie zuvor einfach mal probieren, eine Steckdose zu finden, an der Ihr Computer nicht gestört wird.

2.2.4 Dateneingabeschaltungen

2.2.4.1 Der Rechner streckt die Fühler aus

Nun folgen einige Schaltungen zur Erfassung von Daten. Auch sie werden in ihrer Grundversion an jeweils eine Portleitung angeschlossen. War aber bei den Ausgabeschaltungen noch eine falsche Programmierung des Datenrichtungsregisters ungefährlich, so muß ab jetzt peinlich genau darauf geachtet werden, daß die Portleitung, an der eine der folgenden Eingabeschaltungen betrieben wird, unbedingt als Eingang programmiert ist. Andernfalls kommt es zur Datenkollision, das heißt, falls der Computer und der angeschlossene Sensor unterschiedliche Spannungspegel auf ein und dieselbe Leitung legen, kann der Ausgangstreiber des I/O-Bausteins im Rechner zerstört werden!

Es ist natürlich möglich, jede beliebige TTL-Schaltung (siehe Kapitel 1) an die Porteingänge anzuschließen, da auch sie mit TTL-Pegeln arbeiten. Wenn diese Schaltung jedoch auf Informationen aus der Umwelt reagieren soll, muß auch sie mit geeigneten Wandlern bestückt sein. Oft ist eine sinnvolle Wandlung nur über größere Schaltungen möglich, da ihrer Natur nach analoge (stufenlose) Größen auf irgendeine Weise in digitale HIGH-LOW-Beziehungen umgesetzt werden müssen. Diese Schaltungen benötigen entsprechende Sensoren und nicht selten auch tiefergehende Kenntnisse aus der Analogtechnik. Tabelle 2.3 zeigt eine Auswahl von meßbaren Größen mit den zugehörigen Erfassungsmöglichkeiten.
Die mehr oder weniger kompliziert digitalisierten Daten können Sie zum Beispiel mittels des vorgestellten Minimalprogramms (Listing 2.1) in den Rechner einlesen. In Kapitel 2.2.7 wird jedoch noch ein komfortableres Programm vorgestellt.

2.2.4.2 Einfacher geht's nicht – Eingabe per Schalter

Fangen wir beim Einfachsten an. Bild 2.15 zeigt, wie ein Porteingang per Hand zwischen HIGH und LOW umgeschaltet werden kann. Ist der Schalter offen, liegt die

Meßgröße	geeigneter Wandler
Helligkeit	Fotodiode, Fototransistor, Fotowiderstand
Temperatur	Heißleiter (NTC = Negative Temperature Coefficient), Kaltleiter (PTC = Positive Temperature Coefficient), Bimetallschalter
Spannung	Analog/Digital-Wandler
Strom	Zurückführung auf Spannungsmessung durch Widerstand
Schall	Dynamisches, Elektret-, Kondensator- oder Piezokristallmikrofon
Wegstrecke	Dehnungsmeßstreifen (bis 1mm), optische Abtastung, Ultraschall-Laufzeitmessung
Drehzahl	Tachogenerator, magnetischer Sensor, optische Abtastung, Nockenkontakt
Magnetfeld	Hall-Generator, magnetfeldabhängiger Widerstand
Luftfeuchte	Kapazitiver Feuchtigkeitssensor
Gasdruck	Silizium-Brückensensor, Druckmeßdose

Tabelle 2.3: Wandler zur Erfassung von Meßgrößen

Betriebsspannung von +5 Volt über den sogenannten 'Pull-Up-Widerstand' am Port. Weil nur ein sehr geringer Strom fließt, fällt am Widerstand so gut wie keine Spannung ab. Schließt man den Schalter, so liegt der Porteingang direkt auf Massepotential. Am Widerstand fällt zwar die gesamte Betriebsspannung ab, doch der fließende Strom wird durch den verhältnismäßig großen Wert von 5600 Ohm ausreichend begrenzt. Dabei ist der Wert des Pull-Up-Widerstands unkritisch. Er kann etwa zwischen 500 Ohm und 10 Kiloohm liegen.

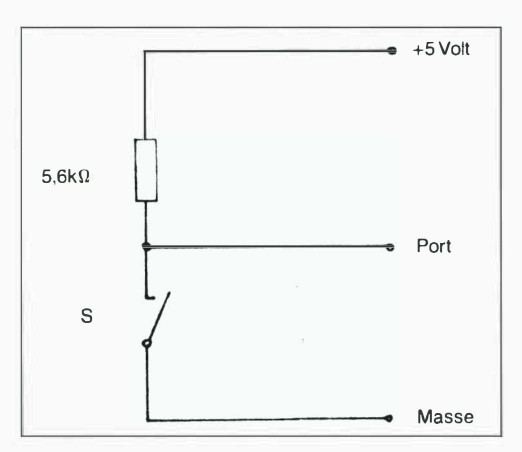


Bild 2.15: Einfachste Dateneingabe über einen Schalter

2.2.4.2.1 Zum Beispiel die Tastatur

Von der gerade beschriebenen einfachen Schaltung wird bereits im C64 intensiv Gebrauch gemacht, nämlich bei der Tastatur. Sie besteht aus einer Matrix von 8 x 8-Tastern, die nach Bild 2.16 den Portleitungen des CIA # 1 zugeordnet sind. Alle Anschlüsse von Port B sind komplett als Eingänge programmiert und bleiben offen, wenn keine Taste gedrückt ist. Andernfalls schließt ein Kontakt die betreffende Leitung von Port B und den jeweiligen Anschluß von Port A kurz. Dann übernimmt der Eingang den logischen Zustand des Ports A, dessen Anschlüsse durchweg als Ausgänge programmiert sind. Im Normalfall führen also alle Bits von Port A HIGH-Pegel.

Zur Tastaturabfrage legt der Computer ebenfalls zunächst an alle Matrixzeilen

HIGH-Pegel an, und schiebt dann ein Null-Bit durch die Zeilen. Wenn er an einem Spalteneingang irgendwann während dieses Vorgangs ein Nullsignal feststellt, dann muß eine leitende Verbindung bestehen, also die entsprechende Taste gedrückt sein. Der zu diesem Zeitpunkt in den beiden Port-Datenregistern stehende Code gibt die Position der gedrückten Taste in der Matrix an. Mit Hilfe einer Tabelle kann das entsprechende 8-Bit-Zeichen herausgefunden werden.

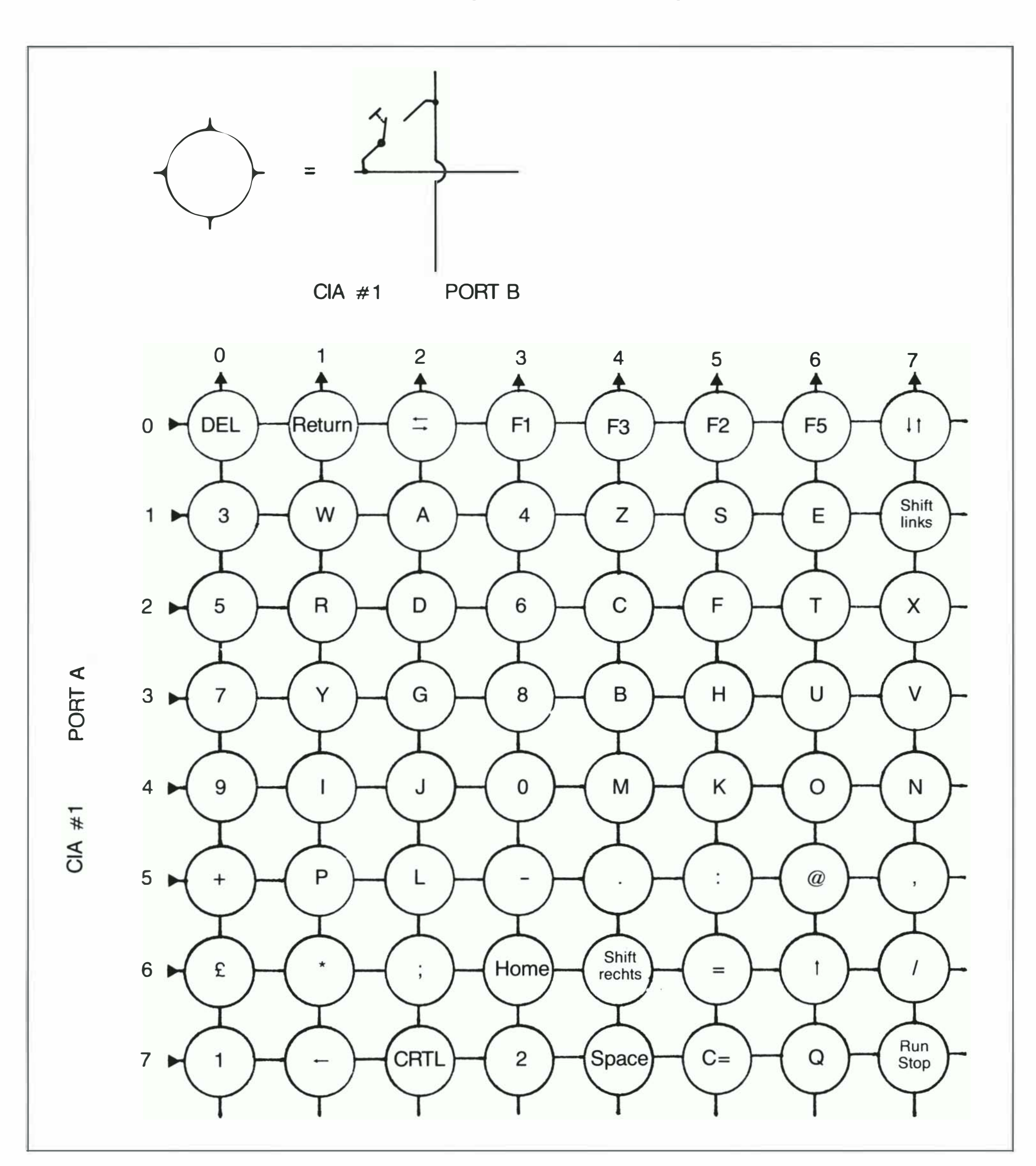


Bild 2.16: Die Position der Tasten in der Matrix

Interessant ist, daß die SHIFT-Tasten unterschiedliche Matrix-Positionen belegen und getrennt abgefragt werden können. Commodore hat sich bei der Entwicklung des C64 darauf verlassen, daß die im Portbaustein integrierten Pull-Up-Widerstände für die Tastaturabfrage ausreichen. Im normalen Betrieb trifft das zu, doch verlängert man das Zuleitungskabel, um eine abgesetzte Tastatur zu realisieren, kann es zu Schwierigkeiten kommen. Es empfiehlt sich dann, zusätzliche Pull-Up-Widerstände (etwa 3,3 Kiloohm) zwischen jede Leitung von Port B des CIA # 1 und +5 Volt einzulöten, am besten bei der Tastatur.

Die Einordnung aller 64 Tasten in die Matrix ist Bild 2.16 zu entnehmen. Im Tastenfeld des C64 befinden sich jedoch noch zwei zusätzliche Funktionen. Zunächst ist da der Schalter SHIFT LOCK, mit dem die linke SHIFT-Taste dauerhaft geschlossen werden kann. Eine absolute Sonderrolle spielt aber die RESTORE-Taste, die über ein Monoflop als Impulsgenerator direkt auf den NMI-Eingang des Prozessors wirkt.

Einzelne Taster werden durch das Bedienfeld des Computers keineswegs überflüssig gemacht, wie es auf den ersten Blick scheinen mag, denn in vielen von Computern gesteuerten Geräten muß ja auf irgendeine Weise die Position von Hebeln, Greifern o. ä. an den Rechner zurückgemeldet werden. Anwendungen wären beispielsweise Endanschlagsmelder mit Springkontakten. Statt des Tasters können in anderen Fällen auch Quecksilber- oder Bimetallschalter eingesetzt werden.

2.2.4.3 Schalten per Licht

Die Eingabeschaltung nach Bild 2.15 gibt bereits eine Grundschaltung der digitalen Datenerfassung an. Das gleiche Prinzip mit Spannungsteiler taucht immer wieder auf. In der Anordnung nach Bild 2.17

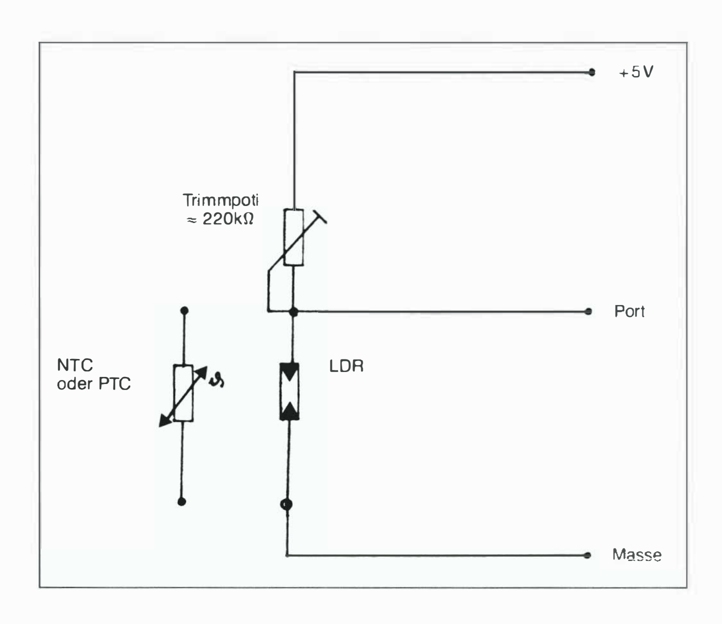


Bild 2.17: Licht oder Temperatur steuert den Computer

wird bei genauerem Hinsehen der Schalter eigentlich nur durch den lichtabhängigen Widerstand (LDR) bzw. durch die temperaturabhängigen Meßfühler NTC oder PTC ersetzt. Die beiden Bauelemente bilden in jedem Fall einen Spannungsteiler, der ähnlich funktioniert wie eben schon erläutert. Der Pull-Up-Widerstand wurde diesmal einstellbar gemacht, um die Empfindlichkeit regeln zu können.

Die angegebene Schaltung kann prinzipiell als Dämmerungsschalter benutzt werden, zum Beispiel um beim Unterschreiten einer gewissen Helligkeit die Beleuchtung einschalten zu lassen. In Kapitel 1.3.2 wurde jedoch bereits gesagt, daß man in der TTL-Technik den Spannungsbereich zwischen den definierten Eingangszuständen meiden soll, da diese Bausteine speziell für steile Flanken ausgelegt sind. Es empfiehlt sich also, einen Baustein vorzuschalten, der am Ausgang schlagartig durchschaltet, wenn am Eingang eine bestimmte Schwelle unter- bzw. überschritten wird. Genau dieses Verhalten zeigt der Schmitt-Trigger, eine bestimmte Schaltvariante, die auch in einzelnen Bausteinen der 74XX-Reihe zu finden ist, beispielsweise im 7413 mit zwei NAND-Gattern zu je vier Eingängen.

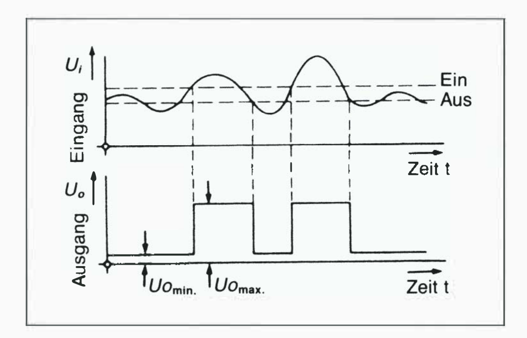


Bild 2.18: Verhalten einer Schmitt-Trigger-Schaltung

Bild 2.18 zeigt den Verlauf der Ausgangsspannung in Abhängigkeit vom Eingang. Man erkennt, daß der Ausgang bei einer etwas höheren Spannung von LOW nach HIGH schaltet, als im umgekehrten Fall von HIGH nach LOW. Der Unterschied zwischen den beiden Spannungen beträgt hier etwa 0,8 Volt. Er wird Hysterese genannt und verhindert Störungen, wie zum Beispiel ständiges Flackern in einem gewissen Übergangsbereich beim Dämmerungsschalter.

Mit einem LDR, einem Widerstand und beispielsweise einer Taschenlampe ist schon der Aufbau einer Lichtschranke möglich. Dabei tritt eine Änderung des Ausgangszustandes auf, wenn die Lichtschranke unterbrochen wird. Lichtschranken sind sehr vielseitige und nützliche Einrichtungen, wie wir noch sehen werden. Ich darf Sie schon jetzt auf die MultiI/O-Platine aus Kapitel 2.2.6 hinweisen, bei der die gerade beschriebene Schaltung mitsamt dem Schmitt-Trigger zum Einsatz kommt.

2.2.4.4 Ein Lichtschranken-Modul

Lichtabhängige Widerstände sind billig und universell einsetzbar. Sie haben jedoch einen gravierenden Nachteil: Sobald die Lichtverhältnisse sehr schnell wechseln, kommen sie mit ihrer Widerstandsänderung einfach nicht mehr nach. Sie reagieren zu träge. Kurze Dunkelphasen werden verschluckt, und mit der Empfindlichkeit steht es auch nicht gerade zum besten.

In kritischen Fällen verwendet man daher gewöhnlich Fotodioden oder Fototransistoren.

Bild 2.19: Schnell reagierender Lichtschrankenempfänger

Die Schaltung nach Bild 2.19 benutzt einen preiswerten Fototransistor, der um so besser leitet, je mehr Licht auf ihn fällt. Man kann sie als Lichtschrankensensor für schnelle Vorgänge einsetzen. Auch hier bildet der Fototransistor zusammen mit seinem Vorwiderstand und dem Trimmpoti einen Spannungsteiler, der im beleuchteten Zustand so eingestellt wird, daß der folgende Schalttransistor T_1 gerade leitet. An seinem Kollektor liegen daher praktisch 0 Volt, so daß T_2 gesperrt ist und am Ausgang HIGH anliegt.

Beim Abdunkeln der Beleuchtung erhöht sich der Innenwiderstand des Fototransistors. Die Basisspannung des Transistors T_1 sinkt also, und am Kollektor von T_1 steht nun relativ hohe Spannung. Damit schaltet T_2 durch, und der Ausgang wird LOW. Sein Zustand entspricht damit der Helligkeit am Fototransistor: Bei hell führt er HIGH, bei dunkel LOW.

Da eine schnelle Lichtschranke bei vielerlei Anwendungen gute Dienste leisten kann, soll die Schaltung hier als kleiner Modul realisiert werden. Sie findet auf der Mini-Platine nach Bild 2.20 Platz, die gleichzeitig eine Befestigung für den Fototransistor darstellt. Der Modul ist – genau wie der entprellte Schalter aus Kapitel 1.5 – an

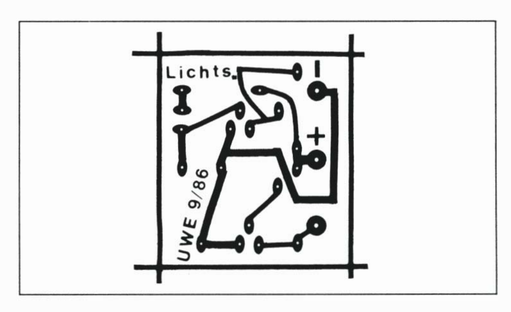


Bild 2.20: Layout des Lichtschrankenmoduls

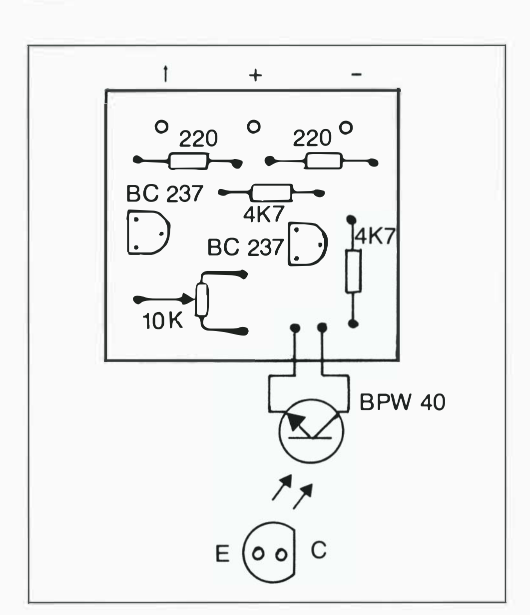


Bild 2.21: Bestückungsplan für das Lichtschrankenmodul

viele der folgenden Platinen über Steckverbindungen direkt anschließbar. Dazu dienen die drei Lötnägel für +5 Volt, Masse und Signal. Die Bauteile (Tabelle 2.4) werden nach dem Bestückungsplan (Bild 2.21) verlötet. Anschließend kann die fertige Schaltung am Zähler aus Kapitel 1 getestet werden. Bei jeder Abdunkelung des beleuchteten Fototransistors muß die Anzeige um eins weiterspringen. Richtig zum Einsatz kommt der Modul beispielsweise in Kapitel 2.4.2.4.

1 Fototransistor	BPW 40
2 Transistoren	BC237
2 Widerstände	220 Ohm
2 Widerstände	4,7 Kiloohm
1 Trimmpoti	10 Kiloohm
3 Lötnägel	
1 einseitige Leiterplatte nach Bild 2.21	

Tabelle 2.4: Die Bauteile für den Lichtschrankenmodul

2.2.4.5 Dateneingabe durch Handauflegen

Mit der Schaltung nach Bild 2.22 ist die Dateneingabe über Sensortasten möglich. Drei hintereinandergeschaltete Transistoren besitzen eine so große Verstärkung, daß bereits über den Hautwiderstand zwischen den beiden offenen Kontakten am Sensortaster S der Schaltvorgang ausgelöst wird. Die Verbindung der Sensortaste

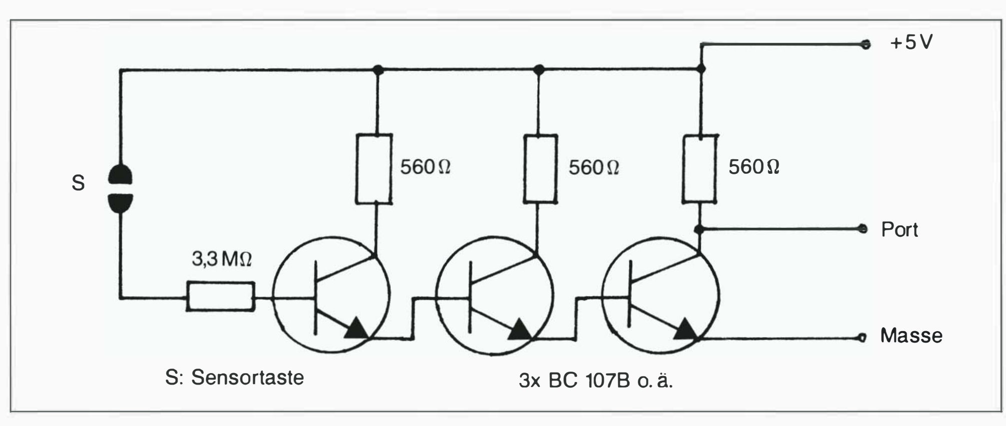


Bild 2.22: Dateneingabeschaltung über Sensortaste

zur Betriebsspannung +5 Volt kann man auch weglassen, da der menschliche Körper als Antenne wirkt und aufgrund der überall vorhandenen Felder von Starkstromnetzen bei Berührung eine ausreichend große Brummspannung an die Basis des ersten Transistors abgibt. Der hohe Basisvorwiderstand dient als Strombegrenzer für den Fall, daß die Sensortaste irrtümlich einmal niederohmig – zum Beispiel durch eine leitende Verbindung – überbrückt wird.

2.2.4.6 Der Computer bekommt Ohren

Einen besonderen Leckerbissen zeigt Bild 2.23. Mit einem Mikrofon kann der Computer sogar auf Schall reagieren. Zunächst bietet die Anordnung sicherlich einen für passionierte Elektroniker ungewöhnlichen Anblick: Da ist ein Inverter als Verstärker zweckentfremdet! Über den 1,7-Megaohm-Widerstand wird dazu sein Ausgang auf den Eingang zurückgekoppelt, so daß sich ein Arbeitspunkt knapp unterhalb der Umschaltspannung einstellt. Ähnlich wie bei einem Operationsverstärker ist dieser Widerstand übrigens für die Empfindlichkeit der Schaltung verantwortlich. Vergrößert man ihn, führen auch leisere Geräusche zur Auslösung. Natürlich wächst damit aber die Störanfälligkeit durch unbeabsichtigte Umwelteinflüsse.

Bei Geräuschen kommt vom Mikrofon eine Spannung, die ausreicht, um am Ausgang ein starkes Signal zu erzeugen. Die positiven Spitzen werden über die Diode an die nachfolgenden Gatter weitergegeben, die sie als eindeutige logische Informationen an den Port durchschalten.

Als Schallaufnehmer muß unbedingt ein Kristallmikrofon verwendet werden, da eine andere, niederohmige Signalquelle den Eingang sofort auf eine zu kleine Spannung herabziehen würde.

Einen Nachteil hat die Schaltung aber noch: Bei manchen Geräuschen sind die Ausgangsimpulse sehr schmal und könnten unter Umständen, wenn das Programm nicht gerade zur richtigen Zeit den Port abfragt, unerkannt verlorengehen. Falls beispielsweise ein Telefon klingelt, entsteht bei jedem Anschlagen des Klöppels an die Glocke ein kurzer Impuls an Pin 6 des Inverterbausteins. Das kann der Computer so nicht auswerten. Mit Hilfe eines nachgeschalteten Monoflops aus Bild 2.24 aber wird daraus für jedes Klingeln ein

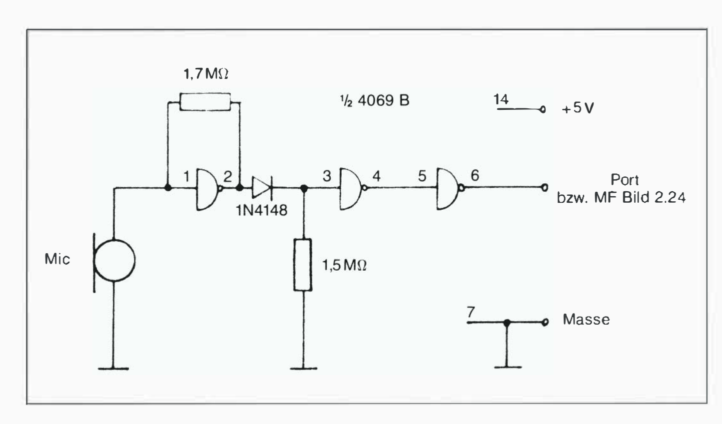


Bild 2.23: Steuerung durch Schall

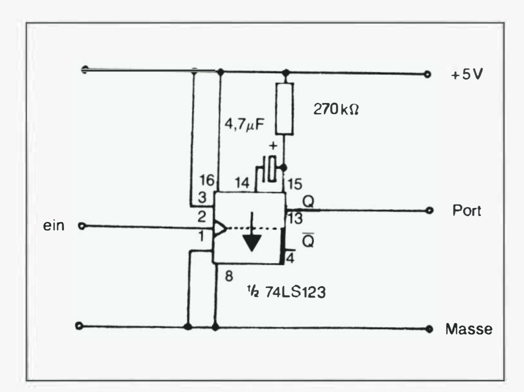


Bild 2.24: Ergänzung zu Bild 2.23: ein Impulsverlängerer

einziger Impuls geformt, denn beim 74123 handelt es sich um ein retriggerbares Monoflop. Das heißt, beim ersten LOW-HIGH-Übergang am Eingang geht das Ausgangssignal für eine bestimmte, mit R und C einstellbare Zeitspanne auf HIGH. Erfolgt während dieser Zeit ein weiterer LOW-HIGH-Übergang, dann verlängert sich die Ausgangs-HIGH-Zeit des Monoflops entsprechend.
In unserem Beispiel ist diese Ausgangszeit mit der RC-Kombination auf etwa 0,6 Sekunden eingestellt. Allgemein errechnet sich die Impulslänge für LS-Typen zu

$$T_w = 0,45 * R * C .$$

Dabei müssen R in Ohm und C in Farad angegeben werden; die Zeit T_w ergibt sich dann in Sekunden.
Anstelle des Mikrofons kann in die Schaltung nach Bild 2.23 auch ein Piezoelement eingesetzt werden, wie es für Alarmanlagen als Glasbruchmelder an Fensterscheiben verwendet wird. Ähnlich dem Kristallmikrofon gibt dieser Sensor bei plötzlicher mechanischer Belastung einen geringen Strom ab, der dann zum Durchschalten der Gatter führt.

2.2.4.7 Komparatoren

Komparatoren sind Spannungsvergleicher und werden vielfach mit Operationsverstärkern aufgebaut, die heute äußerst preisgünstig als integrierte Schaltungen zu haben sind. Operationsverstärker haben unbeschaltet eine extrem hohe Verstärkung und besitzen einen positiven und einen negativen Eingang. Verstärkt wird nur die Differenz zwischen beiden Eingangsspannungen. Betreibt man einen solchen Operationsverstärker nun mit 5 Volt und legt den ersten Eingang auf ein festes Vergleichspotential, den anderen aber an eine unbekannte Spannung, dann sind zwei grundsätzliche Fälle zu unterscheiden:

1. Der zweite Eingang führt höhere Spannung als der Referenzeingang. In diesem Fall ist die Differenz positiv. Der Operationsverstärker nimmt wegen seiner großen Verstärkung schon bei der kleinsten Überschreitung der Referenzspannung seine maximale Ausgangsspannung an. Diese liegt in der Größenordnung der Betriebsspannung, also etwa +5 Volt. Damit ist der Komparatorausgang in diesem Fall HIGH.
2. Der andere Eingang führt niedrigere Spannung als der Referenzeingang. Die Differenzspannung ist nun negativ und der Operationsverstärkerausgang geht sofort auf die kleinstmögliche Spannung. Bei der angegebenen Beschaltung ist dies Massepotential mit 0 Volt, also LOW.

Bild 2.25 zeigt eine reale Beschaltungsmöglichkeit des Komparators. Das Poti (regelbarer Widerstand) dient hier als Spannungsteiler. Man kann es sich ersetzt denken durch zwei hintereinandergeschaltete Einzelwiderstände, zwischen denen der

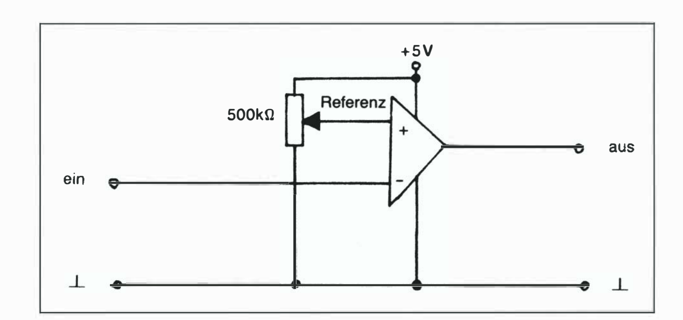


Bild 2.25: Ein Operationsverstärker als Komparator

Mittelabgriff herausgeführt ist. Über das Verhältnis der beiden Widerstände kann man am Abgriff jede beliebige Spannung zwischen der Betriebsspannung (hier +5 Volt) und Masse (0 Volt) einstellen. Der Komparator vergleicht dann diese feste Spannung mit der angelegten Eingangsspannung.

Falls die Referenzspannung nicht einstellbar zu sein braucht, sondern immer einen hochkonstanten Wert haben soll, ist ihre Erzeugung mit Hilfe einer Zenerdiode (auch Z-Diode genannt) angebracht. Bild 2.26 zeigt, wie der Spannungsteiler dann aussehen muß. Die Z-Diode fungiert als spannungsabhängiger Widerstand, der sich stets so einstellt, daß an diesem Bauelement die Z-Spannung abfällt, unabhängig davon, wie hoch die Betriebsspannung des Teilers ist. An der Z-Diode kann also eine stabilisierte Spannung abgegriffen werden. Natürlich hört die Stabilisierungswirkung aber auf, wenn die Betriebsspannung unter den Wert der Z-Spannung sinkt. Es hat also keinen Sinn, eine Z-Diode mit beispielsweise 5,1 Volt in den Teiler einzubauen, wenn die Gesamtspannung nur 5 Volt beträgt. Weil der Komparator einen sehr großen Eingangswiderstand aufweist, kann der Stromfluß durch ihn getrost vernachlässigt werden. Der Vorwiderstand ist daher in weiten Grenzen wählbar. Z-Dioden gibt es in sehr vielen unterschiedlichen Spannungen. Will man ganz bestimmte Werte erreichen, kann man auch mehrere Z-Dioden in Reihe schalten. Die Z-Spannungen addieren sich dann.

2.2.5 Damit Sie wissen, woher der Wind weht

Alle bisher in diesem Kapitel beschriebenen Schaltungen können nur zwei Zustände unterscheiden. Der Taster kann ge-

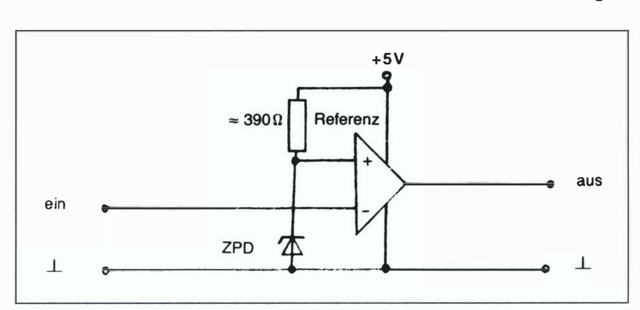


Bild 2.26: Vergleich mit einer festen Spannung

drückt sein oder nicht, der Sensor berührt oder nicht und die Lichtschranke unterbrochen sein oder nicht. Abstufungen sind nicht möglich.

Aus Kapitel 1.7 wissen wir, daß in digitalen Schaltungen auch mehr als zwei Zustände unterschieden werden können, wenn man mehrere Leitungen zusammenfaßt und ihnen unterschiedliche Wertigkeiten zuordnet.

Als Beispiel war dort ein Zähler mit 16 Zuständen auf vier Ausgangsleitungen angegeben. In Bild 2.27 wurden diese 16 Zustände auf vier konzentrischen Kreisen einer Scheibe eingetragen. Beim Ausgangszustand HIGH wurde das entsprechende Feld geschwärzt, bei LOW weiß gelassen. Q_1 bildet dabei den äußersten Ring, Q_4 den innersten.

Denken Sie sich nun auf einer Linie vier lichtabhängige Widerstände (LDRs), die jeweils über einen Ring angebracht sind. Von unten wird die Scheibe durchleuchtet, es entstehen also vier Lichtschranken. Wenn sich unter einem LDR ein geschwärztes Feld befindet, ist die Lichtschranke unterbrochen und der entsprechende Kanal LOW, andernfalls wird HIGH ausgegeben.

Je nachdem wie man die Scheibe dreht, ergeben sich an den Lichtschranken wieder die 16 Ausgangszustände aus Bild 1.24. Wir haben also ein Gerät entwickelt, das den Drehwinkel der Scheibe anzeigt.

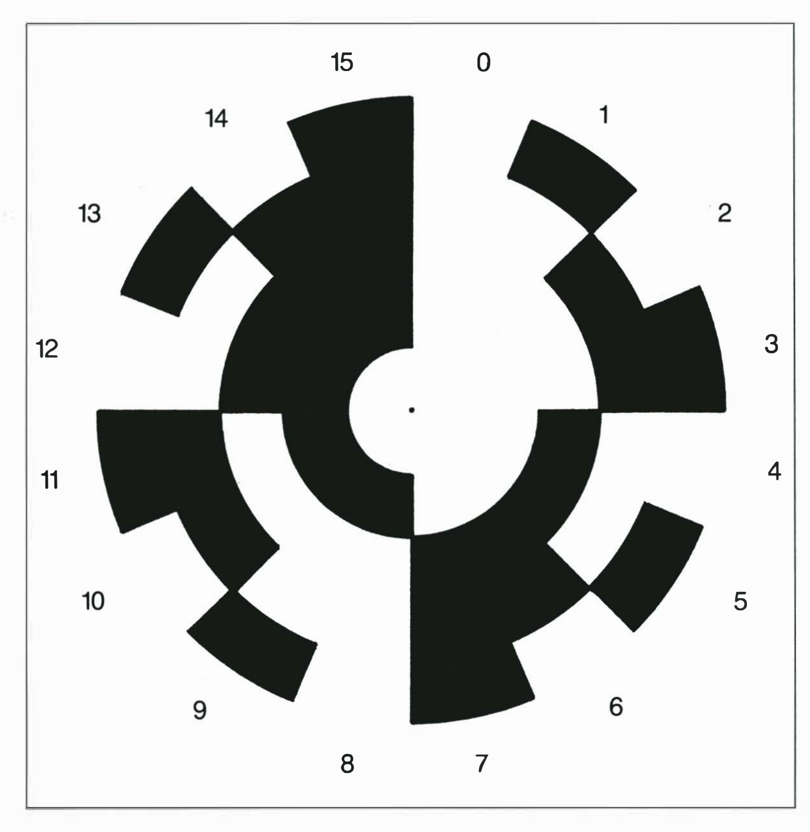


Bild 2.27: Code-Scheibe mit dualer Einteilung

Hier sieht man deutlich, daß der Meßwert nicht analog, sondern digitalisiert vorliegt. Nimmt man die Grenze zwischen 0 und 15 als 0 Grad, dann ergibt sich eine Zuordnung nach Tabelle 2.5.

Anzeige	Gradzahl
0	0 – 22,5
1	22,5 – 45
2	45 – 67,5
3	67,5 – 90
4	90 – 112,5
5	112,5 – 135
6	135 – 157,5
7	157,5 – 180
8	180 – 202,5
9	202,5 – 225
10	225 – 247,5
11	247,5 – 270
12	270 – 292,5
13	292,5 – 315
14	315 – 337,5
15	337,5 – 360

Tabelle 2.5: Zuordnung der Anzeige zum realen Winkel

2.2.6 Eine Multi-I/O-Platine

Damit Sie die Ausführungen praktisch nachvollziehen können, folgt jetzt die Beschreibung einer Multifunktionsplatine zur Ein- und Ausgabe. Sie enthält mehrere bereits beschriebene Schaltungen.

Da wären zunächst vier Sensoreingänge mit einstellbarer Empfindlichkeit und nachgeschaltetem Schmitt-Trigger, die beispielsweise für die genannten Versuche mit dem Windrichtungsanzeiger wichtig sind.

Die übrigen vier Kanäle wurden mit Ausgabeschaltungen belegt, und zwar mit der 2-Bit-Motorsteuerung nach Bild 2.3 sowie mit zwei Open-Collektor-Universalausgängen, deren Zustand jeweils über eine Leuchtdiode angezeigt wird (siehe Kapitel 2.2.3.2).

Zu Experimentierzwecken ist auch der Anschluß $\overline{PC2}$, auf den wir später noch zu sprechen kommen werden, gepuffert an einem Lötnagel zugänglich.

Der Aufbau dieser Karte erfolgt wie gewöhnlich nach dem Bestückungsplan

Anzahl	Bauteil
	Grundausbau:
1	User-Port-Stecker
1	Kondensator 10 Mikrofarad / 10 Volt
1	Platine nach Bild 2.31
	Eingangsstufen und $\overline{PC2}$-Pufferung:
1	7414
1	Sockel 14pol
4	Trimmer 250 Kiloohm liegend (kleine Bauform)
1	Widerstand 3,3 Kiloohm
13	Lötnägel
	Ausgänge Grundausbau:
1	7407
1	Sockel 14pol
	Motorsteuerung:
4	Dioden 1N4004
2	Transistoren BD137 (NPN)
2	Transistoren BD138 (PNP)
2	Widerstände 1 Kiloohm
4	Lötnägel
	Universalausgänge:
2	Leuchtdioden 3mm
2	Widerstände 330 Ohm
4	Lötnägel

Tabelle 2.6: Bauteile für die Multi I/O-Karte

(Bild 2.30 Seite 74) und den allgemeinen Hinweisen in Anhang A. Tabelle 2.6 zeigt eine Zusammenstellung der benötigten Bauelemente. Natürlich müssen nicht alle Schaltungsteile gleichzeitig aufgebaut werden, sondern die einzelnen Funktionseinheiten lassen sich nach und nach – je nach Interesse oder Geldbeutel – vervollständigen. Daher wurde die Bauteilliste in Abschnitte unterteilt. Auf jeden Fall brauchen Sie die ganz zu Anfang stehenden Elemente. Wollen Sie nur einen Teil der Ausgänge beschalten, benötigen Sie immer auch das IC 7407.

Der User-Port-Stecker wird mit seiner unteren Kontaktleiste auf die Platinenunterseite gelötet. Zu der oberen Leiste führen lediglich zwei Drahtbrücken.

Foto 2.1 zeigt die vollständig bestückte Platine.

Hierzu noch eine kurze Übersicht zur Funktion der einzelnen Bits in Tabelle 2.7.

Baut man die Code-Scheibe nach Bild 2.28 mit etwas handwerklichem Geschick in die Anordnung nach Bild 2.29 ein, dann ist es möglich, mit dem Computer die Windrichtung zu messen. Die LDRs erhalten Blenden, die nur Licht aus Richtung der Scheibe einfallen lassen. Damit werden Störungen durch Streulicht ausgeschaltet. Für eine dauerhafte Installation im Freien sollte das Gehäuse wasser- und lichtdicht sein.

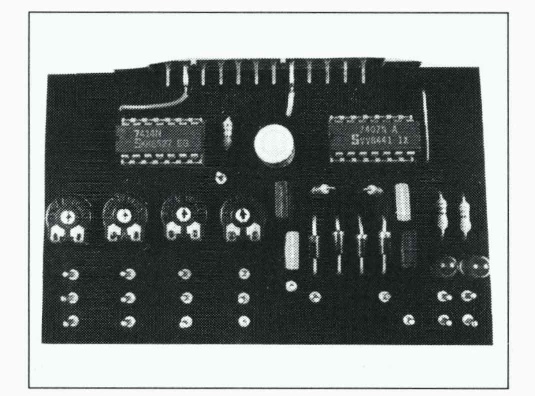


Foto 2.1: Die fertige Multi-I/O-Platine

2.2.7 Portsupport

Beim Experimentieren ist es oft recht umständlich, einzelne Bits gezielt zu ändern, indem man mit Wertigkeiten hantiert und vor jedem POKE-Befehl endlose Additionen im Kopf durchführt. Der Computer macht es uns einfacher.

Das Programm PORTSUPPORT besteht

Bit	Wertigkeit	Richtung	Funktion
0	1	zum C64	Sensor 0
1	2	zum C64	Sensor 1
2	4	zum C64	Sensor 2
3	8	zum C64	Sensor 3
4	16	vom C64	Motor
5	32	vom C64	Motor
6	64	vom C64	Universalausgang
7	128	vom C64	Universalausgang

Tabelle 2.7: Funktionen der Multi-I/O-Platine

Bild 2.28: Kodierung nach dem Gray-Code

Bild 2.29: Anordnung zum Messen der Windrichtung

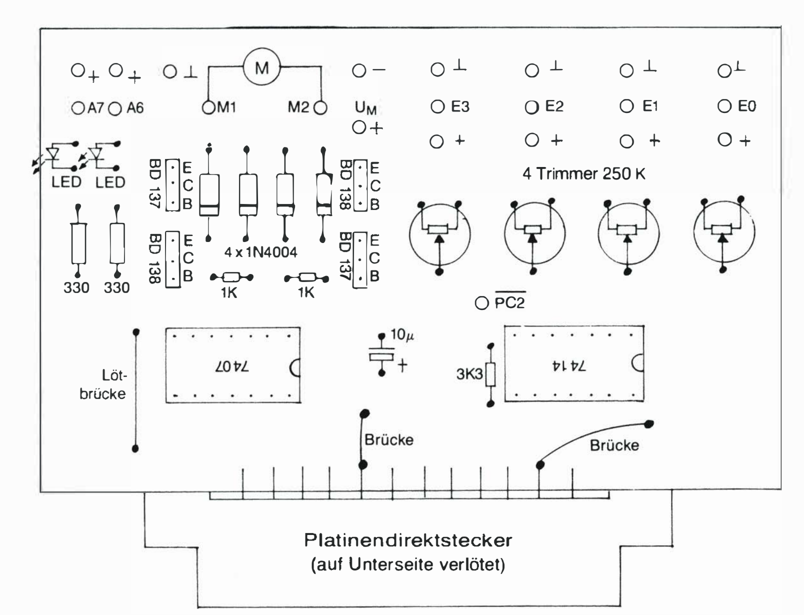


Bild 2.30: Bestückung der Multi-I/O-Platine

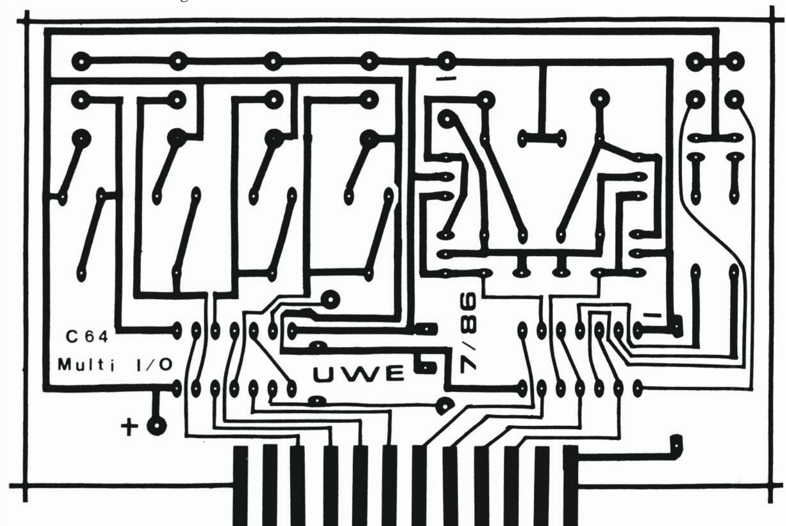


Bild 2.31: Layout für die Multi-I/O-Platine

aus mehreren Unterprogrammen, die wenigstens die Rechnungen übernehmen und auch in eigene Programme einbaubar sind. Das Hauptprogramm gibt ein Beispiel zur Benutzung, anhand dessen alle Einzelfunktionen vorgestellt werden sollen:

Im ersten Unterprogramm ab Zeile 9000 (Initialisierung) kann man bequem jeden der acht verfügbaren Anschlüsse von

Port B als Eingang bzw. Ausgang definieren. Eine Schleife bearbeitet jeden Kanal. Dabei genügt die Angabe von E oder A nach der jeweiligen Frage. Falls A getippt wird, addiert das Programm die Wertigkeit zur Variablen D, die anschließend ins Datenrichtungsregister gePOKEt wird. Sicherheitshalber führt jede andere Eingabe zur Programmierung als Eingang.

Es folgt der Aufruf einer ganz ähnlichen Routine zur bitweisen Festlegung des Ausgabewertes. Hier muß – beginnend mit dem niederwertigsten Bit 0 – angegeben werden, welche Stellen gesetzt (1) und welche nicht gesetzt (0) sein sollen. Das Ergebnis wird in der Variablen W festgehalten.

Erst bei Aufruf der Routine AUSGABE, erfolgt die Änderung des Portregisters, indem W hineingeschrieben wird.

Den gegenteiligen Fall besorgt die Unterroutine PORT LESEN ab Zeile 9200. Dort wird der Port ausgelesen und sein Inhalt in W zur Verfügung gestellt. Diese beiden Routinen bestehen jeweils nur aus einem einzigen Befehl und wären eigentlich gar kein Unterprogramm wert.

Interessanter ist da schon der Schluß: eine Dezimal-Dual-Wandlung. Der Inhalt der Variablen W wird durch sie als Bitkombination angezeigt, was schnelle Aussagen über den Zustand der einzelnen Bits zuläßt. Die linke Stelle steht für das höchstwertige Bit 7, die ganz rechts für Bit 0.

Im Hauptprogramm beginnt mit Zeile 140 eine Schleife, in der bei jedem Durchlauf der Port gelesen und sein Inhalt zuerst dezimal (Zeile 150) und nach Wandlung durch das Unterprogramm ab Zeile 9900 auch dual angezeigt wird (Zeile 170).

Damit eignet sich dieses kurze Programm sehr gut für Test- und Experimentierzwecke.

Beim Betrieb mit der Multi-I/O-Karte müssen die Kanäle 0 bis 3 als Eingänge, 4 bis 7 jedoch als Ausgänge programmiert werden. Beim Experimentieren werden Sie sehen, daß beim Lesen der als Ausgang programmierten Kanäle immer der zuletzt hineingeschriebene Wert ausgegeben wird.

2.2.8 Maskerade – Einige Softwaretips

Gerade bei Hardware-Programmierung kommt es sehr oft vor, daß jedes Bit einer Speicherzelle eine andere Bedeutung hat. Um ganz bestimmte Reaktionen hervorzurufen, wird es dann nötig, einzelne Bits gezielt setzen und löschen zu können, ohne daß der übrige Inhalt der Speicherzelle auch nur kurzzeitig geändert wird.

Dem gerade vorgestellten Programm muß für jede Stelle der Zustand angegeben werden. Oft weiß man diese übrigen Werte aber gar nicht, sondern man will nur das eine Bit ändern.

Zu diesem Zweck gibt es einen Programmiertrick, der sowohl bei Assembler als auch unter BASIC funktioniert. Beide Sprachen verfügen nämlich über die logischen Befehle AND und OR. Dabei werden die Operanden jeweils bitweise miteinander verknüpft. Leider arbeitet BASIC nur mit Dezimalzahlen, was die Anschaulichkeit etwas vermindert, aber in Kapitel 1.7.2 haben Sie ja bereits die Umrechnung kennengelernt.

Was bedeutet nun bitweise Verknüpfung? Gehen wir einfach von einem Beispiel aus. Der BASIC-Befehl

```
2 OR 8
```

führt eine ODER-Operation zwischen 2 und 8 durch. Binär sehen diese Operanden folgendermaßen aus:

```
00000010   2
00001000   8 .
```

Jeweils die untereinanderstehenden Bits werden nun verknüpft. Es ergibt sich

```
00001010   10 .
```

Auf diese Weise wurde also das Bit 3 zusätzlich gesetzt. Das funktioniert immer, egal welchen Wert der erste Operand hat. Mit

```
A = A OR 8
```

kann man einen Wert lesen, die Verknüpfung durchführen und den modifizierten Wert wieder zurückschreiben.
Man nennt den zweiten Wert auch Maske. A wird also in diesem Fall mit 8 maskiert.
Ähnlich verläuft die Sache beim Löschen einzelner Bits. Diesmal wird lediglich die Maske und der ursprüngliche Speicherinhalt UND-verknüpft. Die zu löschende Stelle muß in der Maske ebenfalls gelöscht sein, denn eine UND-Verknüpfung mit 0 ergibt ja wieder 0. Alle anderen Stellen müssen dagegen gesetzt, also 1 sein. Will man das gesetzte Bit des obigen Beispiels wieder löschen, dann lautet die Maske

```
11110111 , also  255-8=247
```

und der zugehörige Befehl

```
A = A AND 247 .
```

Hat man diesen Trick einmal verstanden, bereitet es keine Probleme mehr, beliebige Bitmanipulationen auszuführen.
Ein anderes Problem ist die Isolierung von einzelnen Bits. Auch hier hilft der AND-Befehl weiter. In der Maske darf diesmal nur an der interessierenden Stelle eine 1 stehen. Alle anderen Bits sind 0. Nach der Verknüpfung ergibt sich der Wert 0, falls das entsprechende Bit nicht gesetzt war. Andernfalls erhält man die Wertigkeit des gesetzten Bit.
Auch dazu ein Beispiel:
Am User-Port liegt die Bitkombination

```
11011001 ,
```

also 217. Uns interessiert nur das Bit 3. Es hat die Wertigkeit 2 hoch 3 = 8. Damit lautet die Maske

```
00001000.
```

Eine UND-Verknüpfung liefert

```
00001000,
```

also 8. Das Bit war gesetzt. Mit der Portadresse 65577 lautet der entsprechende BASIC-Befehl

```
PRINT PEEK(65577) AND 8 .
```

Ist das Ergebnis 8, war das Bit gesetzt, bei 0 war es dagegen nicht gesetzt.
Listing 2.2 macht ausgiebig Gebrauch von den beschriebenen Techniken. Es wird in Kapitel 2.2.10.3 noch eingehend beschrieben.

2.2.9 Ein Alternativ-Port in der Floppy

Hardware-Bastler werden sich bestimmt schon oft darüber geärgert haben, daß bereits ein einziges angeschlossenes Gerät am User-Port diesen interessanten Verbindungskanal zur Computerumwelt verstopfen kann. Betreibt man beispielsweise – wie im nächsten Abschnitt erläutert wird – seinen Centronics-Drucker über den Par-

allel-Port, so ist es nahezu unmöglich, ohne raffinierte technische Klimmzüge zwischendurch mal eben die Temperatur oder die Beleuchtungsstärke zu kontrollieren, ein Gerät ein- bzw. auszuschalten, die Modelleisenbahnanlage zu steuern oder anderweitig Daten zu übertragen. Doch das ist leicht zu ändern.

Den Besitzern einer Floppy VC 1541 steht bereits die volle Hardware für einen zusätzlichen User-Port zur Verfügung, denn dort befinden sich zwei VIA-Bausteine vom Typ 6522 mit jeweils zwei 8 Bit breiten Ports. Der eine Baustein bedient die Kopf- und Motorsteuerung, während der andere nur zur Datenübertragung vom und zum Computer zuständig ist. Da der serielle Bus aber nicht alle Anschlüsse benötigt, bleibt der komplette Port A des mit UC3 bezeichneten Bausteins unbenutzt und kann für eigene Zwecke eingesetzt werden.

Einziger Schwachpunkt ist, daß die entsprechenden Anschlüsse nicht herausgeführt sind. Hier muß man also wohl oder übel mit dem Lötkolben in das Innenleben des Gerätes eingreifen. Es ist empfehlenswert, an der Seite des Floppygehäuses einen Steckverbinder zu montieren, auf den auch Anschlüsse für Masse und +5 Volt Versorgungsspannung gelegt werden sollten. Die benötigten Lötpunkte am Floppyschaltkreis UC3 sind aus der Tabelle 2.15, Seite 97, zu entnehmen. Es gibt auch die Möglichkeit, mittels eines Flachbandkabels ein Experimentierboard auf der Arbeitsfläche direkt anzuschließen.

Über den seriellen Bus können die Steuer- und Datenregister des floppyinternen I/O-Bausteins mittels der Memory-Read- bzw. Memory-Write-Befehle wie gewöhnliche Speicherzellen im Floppy-RAM angesprochen werden. Vielleicht sollten Sie sich die entsprechenden Befehle des 1541-DOS kurz noch einmal ins Gedächtnis rufen:

Mit dem Memory-Write-Befehl kann man eine Anzahl von Bytes ins RAM der Floppy schreiben. Seine Syntax lautet folgendermaßen:

```
PRINT#1, ''M—W''  CHR$(LO) CHR$(HI)
                  CHR$(Anzahl)
                  CHR$(Data 1)
                  CHR$(Data 2) . . .
```

Dabei stehen LO und HI für niederwertigen bzw. höherwertigen Teil der RAM-Adresse und Anzahl für die Zahl der zu übertragenden Bytes. Anschließend werden die Datenbytes übergeben.

Beim Memory-Read-Befehl wird dagegen die Adresse mit

```
PRINT#1, ''M—R'';CHR$(LO); CHR$(HI)
```

übertragen und deren Wert dann mit

```
GET#1, A$
```

abgeholt. Die zuständigen Adressen für den bisher ungenutzten Port A des 6522 (UC3), sowie die zugehörigen High- und Lowbytewerte zeigt Tabelle 2.8, Seite 78. Genauere Angaben dazu findet man im Markt & Technik-Buch »Die Floppy 1541« von Karsten Schramm.

Das Programm ALTERNATIVPORT entspricht dem Programm PORTSUPPORT aus Kapitel 2.2.7, mit dem Unterschied, daß die Routinen hier über den seriellen Bus auf den Floppy-Port wirken. Sie sind mit den genannten Befehlen aufgebaut und geben dem Anwender ein ausreichendes Instrumentarium zur unkomplizierten

Register	Hexadezimal	Highbyte	Lowbyte
Portregister	$1801	24	01
Datenrichtungsregister	$1803	24	03

Tabelle 2.8: Benutzte Floppy-Adressen

Nutzung des zusätzlichen I/O-Ports an die Hand.
Die Funktionen wurden bereits beschrieben. Für den VIA-Baustein 6522 in der Floppy gelten die gleichen Grundsätze wie für den CIA-Baustein 6526 im C64.

2.2.10 Digitale Computer-Lichtorgel

Heiße Parties bekommen erst durch eine stimmungsvolle Beleuchtung ihren richtigen Pepp. Rhythmisch im Takt der Musik blinkende Lichter müssen her, doch meist reicht bei Jugendlichen das Taschengeld gerade mal für die Party-Utensilien.
Der Lichtorgelzusatz zum C64 sorgt da für Abhilfe. Er ist schnell und vor allem billig aufzubauen, in seiner Grundversion mit einem Mini-BASIC-Programm schon sehr effektvoll und nach Wunsch stufenweise zu einer kompletten Lichtanlage ausbaubar. Die Effektmöglichkeiten sind durch individuelle Programmierung nahezu unbegrenzt.

2.2.10.1 Hardwareankoppelung

Eine Lichtorgel ist im Prinzip ein elektronischer Schalter, der von einem Signalgeber angesteuert wird und normalerweise eine oder mehrere Glühlampen im Rhythmus der Musik aufleuchten läßt. Als Signalquelle kann ein Kofferradio, Tonbandgerät, Plattenspieler oder eine ganze Stereoanlage genommen werden, egal ob mit Netz- oder Batteriebetrieb. Bild 2.32 zeigt die Zusatzschaltung für den C64, die aus zwei Teilen besteht.
In der oberen Hälfte ist der Eingabeteil zu sehen.
Die Ankoppelung ist so gewählt, daß sie die Geräte nur minimal belastet. Vom Lautsprecherausgang – oder vom eingebauten Lautsprecher direkt – wird die Tonfrequenz (NF) einem Übertrager mit dem Übersetzungsverhältnis 1:10 zugeführt, der die Aufgabe hat, den niederohmigen Ausgang des Verstärkers an den hochohmigen Eingang der Zusatzschaltung anzupassen und das Signalgerät galvanisch von der Lichtorgelschaltung zu trennen. Am Ausgang des Übertragers (hochohmige Seite) liegt ein als Spannungsteiler geschaltetes Potentiometer, mit dem die Ansprechempfindlichkeit geregelt werden kann. Danach gelangt das Signal über einen Kondensator zur Basis des ersten Transistors, der es verstärkt und einer Schaltstufe zuführt, deren Ausgang am User-Port-Anschluß PA2 liegt und so vom Computer abgefragt werden kann. Dieser Ausgang führt normalerweise HIGH und wird nur dann LOW, wenn er von der Signalquelle durchgesteuert wird.
Der zweite Kondensator in der Verstärkerstufe ermöglicht es, durch Gegenkoppelung nur bestimmte Frequenzbereiche auszufiltern und zu verstärken. Würde man die Lichtorgelstufe mehrmals mit geänderten Kondensatorwerten aufbauen, dann erhielte man getrennte Kanäle für Höhen,

Bild 2.32: Schaltplan der Lichtorgel

Mitten und Tiefen. Dabei müßten die Potis jeweils parallel zu dem Empfindlichkeitsregler des gezeichneten Kanals liegen.

Der untere Schaltungsteil besteht aus acht getrennt programmierbaren Ausgangstreibern. Sie sind alle zusammen im IC 74LS241 enthalten und liefern genügend Strom, um Leuchtdioden anzusteuern. Damit eignen sie sich zum Betrieb der Lastschaltung für Glühbirnen (Bild 2.10 oder ähnliche). Die Lastschaltungen selbst finden auf der Platine nach dem Layout aus Bild 2.33 keinen Platz, sondern müssen extern in einem isolierten Gehäuse untergebracht werden. Der An-

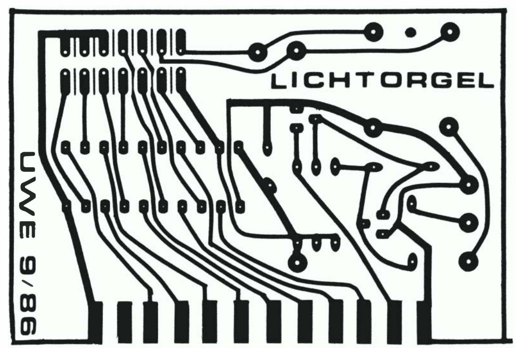


Bild 2.33: Layout für die Lichtorgel-Schaltung

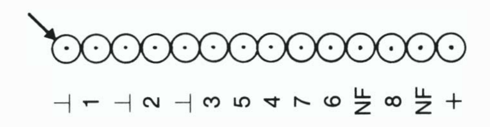


Bild 2.34: Belegung des Flachbandkabels

schluß kann über ein Flachbandkabel erfolgen, das mittels einer Pfostensteckverbindung in Schneidklemmtechnik sehr leicht und schnell herstellbar ist. Bild 2.34 zeigt die Belegung des Flachbandkabels. Die Pfeile dort und an der Pfostenstеckerleiste im Bestückungsplan (Bild 2.35) markieren die Einbaurichtung.

Oberseite (Pin 2). Die Zuführung der NF-Signale vom Verstärker kann wahlweise über das Lüsterklemmenpaar oder durch das Flachbandkabel erfolgen. Der Übertrager muß unbedingt richtig herum eingelötet werden. Er besitzt einen Farbpunkt zur Kennzeichnung der Eingangsseite (dickerer Draht). Dieser Farbpunkt ist sowohl im Bestückungsplan als auch im Platinenlayout deutlich gekennzeichnet.

Wollen Sie keine 220-V-Glühbirnen anschließen, sondern nur die Effekte auf dem Bildschirm sichtbar machen, dann

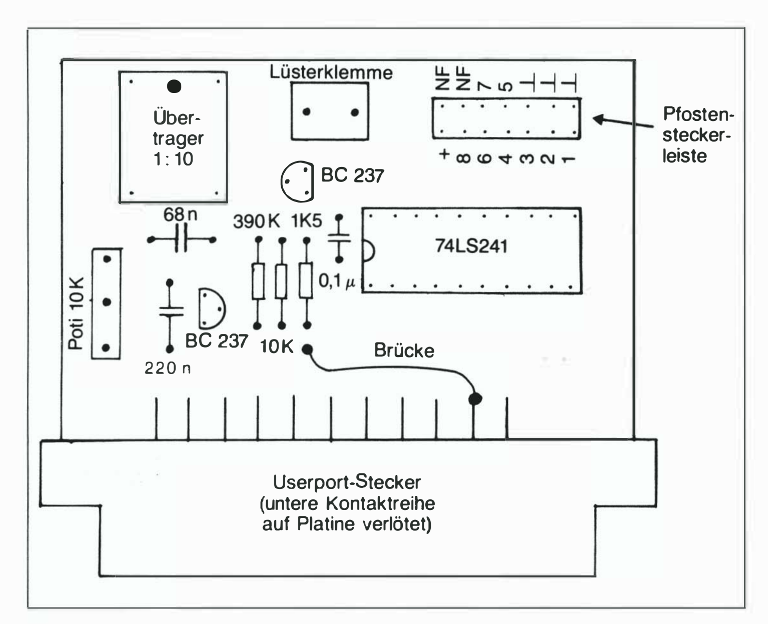


Bild 2.35: Lichtorgel-Bestückung

2.2.10.2 Aufbau der Lichtorgel

Die wenigen Bauteile (Tabelle 2.9) sind schnell nach dem Bestückungsplan (Bild 2.35) verlötet. Dabei wird die Platine mit der unteren Kontaktreihe des User-Port-steckers verbunden. Vergessen Sie nicht die Brücke zum +5V-Anschluß auf der

können natürlich IC und Steckerleiste entfallen.

Andersherum ist es selbstverständlich auch möglich, die Platine nur zum Ansteuern von Leuchtdioden oder ähnlichem per Programm zu benutzen. In diesem Fall braucht der Lichtorgelteil nicht bestückt zu werden.

Ansonsten beachten Sie bitte die Tips zur Platinenherstellung im Anhang A.

	Grundausbau
1	User-Port-Stecker
1	Kondensator 0,1 Mikrofarad, Keramik
1	einseitige Platine nach Bild 2.33
	Lichtorgelteil:
1	Übertrager 1:10
2	Transistoren BC237
1	Widerstand 1,5 Kiloohm
1	Widerstand 10 Kiloohm
1	Widerstand 390 Kiloohm
1	Kondensator 68 Nanofarad
1	Kondensator 220 Nanofarad
1	Poti 10 Kiloohm
1	Lüsterklemmenpaar zum Einlöten
	Lastansteuerung:
1	Bustreiber 74LS241
1	IC-Sockel 20-pol
1	Pfostensteckleiste 2 x 7-pol

Tabelle: 2.9: Bauteile für die Lichtorgelplatine

Foto 2.2 zeigt die fertig aufgebaute Platine, und in Foto 2.3 sieht man ein achtkanaliges Lichtpult, das unter anderem auch

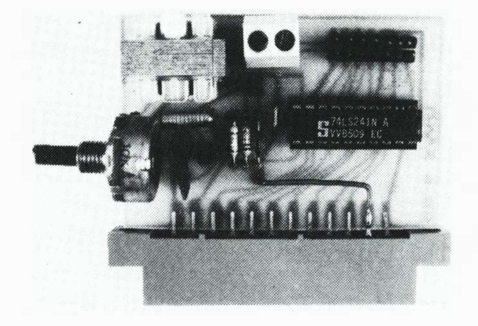


Foto 2.2: Der fertig bestückte Lichtorgelzusatz

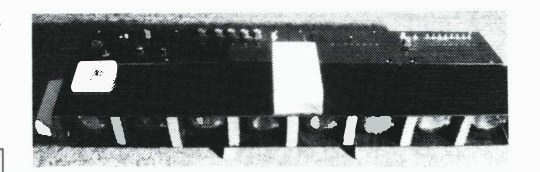


Foto 2.3: Eine komplette Lichtanlage des Autors

den Computer als Signalquelle für die 220-V-Strahler zuläßt.
Mit dem Programm LAUFLICHT.DEMO wird reihum jeder Ausgangskanal kurz angesteuert. Es dient zum Testen der Funktion bei angeschlossener Lastschaltung.

2.2.10.3 Ein Mini-Lichtorgelprogramm

Listing 2.2 (LICHTORGEL) zeigt ein wirkungsvolles Effektprogramm in BASIC, das ausschließlich den Bildschirm benutzt. Es fragt in einer Schleife den Zustand des Lichtorgelausgangs an PA2 ab und ändert die Farben, wenn dort LOW anliegt. Gehen wir einmal das ganze Programm genau durch.
In Zeile 100 wird zunächst der Pin PA2 als Ausgang programmiert. Das geschieht, indem über eine Maske nur das zuständige Bit 2 (Wertigkeit 4) gesetzt wird. Alle übrigen Bits bleiben unverändert. Lesen Sie dazu bitte auch Kapitel 2.2.8 (Maskerade). Anschließend wird der Bildschirm gelöscht, und es beginnt eine Endlosschleife, in der mit zwei gleichartigen Zeilen einmal die Rahmenfarbe und dann die Hintergrundfarbe geändert wird. Dazu fragt das Programm jeweils das Portbit PA2 ab, indem es das gesamte Portregister A liest und mit 4 maskiert. Bei gesetztem PA2-Bit ergibt sich der Wert 4, und es passiert nichts weiter. Ist PA2 jedoch LOW, dann ergibt die logische Verknüpfung 0, und die

```
100 POKE 56578,(PEEK(56578) AND (255-4)): REM PA2 AUF EINGANG
120 :
200 IF (PEEK(56576) AND 4)=0 THEN POKE 53280,(PEEK(53280) AND 15)+1
210 IF (PEEK(56576) AND 4)=0 THEN POKE 53281,(PEEK(53281) AND 15)+1
220 GOTO 200
```

Listing 2.2: Ein Lichtorgel-Effektprogramm

jeweilige Farbe wird geändert, indem sie um 1 erhöht wird. Wie Sie alle wissen, ist der Vorgang in dieser Form jedoch nur 256mal möglich, denn falls man versucht, eine größere Zahl als 255 in eine Speicherzelle zu POKEn, bekommt man eine Fehlermeldung.

Um sie zu vermeiden, wurde ein Trick angewandt, der ausnutzt, daß nur die niederwertigsten 4 Bit in den Farbregistern relevant sind, denn es gibt ja beim C64 nur 16 verschiedene Farben. Die Addition im POKE-Befehl wird daher nur mit diesen 4 Bit durchgeführt, indem sie mittels AND 15 isoliert werden. Ein Beispiel soll das deutlich machen:

	00011001	25
AND	00001111	15
	00001001	9

Im ursprünglichen Wert sei Bit 4 gesetzt. Durch die AND- Verknüpfung mit 15 werden alle gesetzten Bits im höherwertigen Halbbyte ausgeblendet. Bei der anschließenden Addition von 1 können also nie größere Werte als 16 auftreten, da diese Zahl beim nächsten Durchlauf bereits wieder als 0 gewertet würde.

Das kurze Programm sorgt für erstaunlich gute Effekte, die zum Teil noch bessere Wirkung haben, als eine normale Lichtorgel. Natürlich sind auch andere Effekte programmierbar, wie beispielsweise die Ansteuerung der Lastausgänge als taktabhängiges Lauflicht und auf Knopfdruck wechselnde Farben.

Die beiliegende Diskette enthält auch ein Maschinenprogramm, das viel schneller reagiert, als es unter BASIC möglich ist und schon von daher interessanter wirkt. Im Rahmen werden spezielle Grafikmuster erzeugt. Zum Betrieb laden Sie bitte den Object-Code LICHTORGEL.OBJ und starten ihn mit SYS 12*4096. ($C000)

2.2.11 Digital/Analog-Wandlung

Wie man binäre in dezimale Zahlen umwandelt, wurde schon in Kapitel 1.7.2 besprochen. Jede Stelle hat dabei eine bestimmte Wertigkeit, nämlich genau eins mehr als die Summe aller Wertigkeiten darunter. Wie wäre es, wenn man diese Wertigkeiten durch Spannungen darstellen könnte? Jede Bitkombination entspräche dann einer Spannung.

Die hier vorgestellte Schaltung macht das möglich. Auch sie kommt allein mit den 8 Portbit aus.

2.2.11.1 Operationsverstärker

Es gibt in der Computertechnik diverse Versuche, Rechnungen mit analogen Grö-

ßen auszuführen. Statt die kontinuierliche Ausgabespannung eines Meßfühlers erst mehr oder weniger genau in Zahlen umzuwandeln, nimmt man dabei gleich den stufenlosen Wert und verarbeitet ihn unmittelbar weiter. Theoretisch erhält man mit dieser Methode genauere Ergebnisse. Natürlich sind für solche Geräte hochgenaue Verstärker mit nahezu idealen Eigenschaften nötig, die sich allerdings technisch nur sehr schwer realisieren lassen. Eine spezielle Gruppe solcher Analogverstärker sind die Operationsverstärker. Ihren Namen erhielten sie, weil sie zuerst in Analogrechnern zur Ausführung von Rechenoperationen eingesetzt wurden. Dank der integrierten Schaltungstechnik und ihrer universellen Einsetzbarkeit sind sie heute sehr preisgünstig zu haben.

Es gibt verschiedene Ausführungen von Operationsverstärkern. Allen gemeinsam sind jedoch folgende Eigenschaften: Sie besitzen zwei Eingänge (einen invertierenden und einen nicht invertierenden), verstärkt wird die Differenzspannung zwischen diesen Eingängen, und zum Betrieb benötigt man eine positive und eine negative Versorgungsspannung. Die Verstärkung ist über die äußere Beschaltung in weiten Grenzen einstellbar.

Mit Operationsverstärkern kann man allein durch wenige externe Bauelemente analoge Addierer, Subtrahierer, Multiplizierer, Differenzierer und Integrierer realisieren.

In dieser Schaltung soll ein Operationsverstärker als Addierer benutzt werden. Zum Einsatz kommt die bekannte Ausführung 741, die von vielen Herstellern produziert und unter verschiedenen Bezeichnungen angeboten wird, zum Beispiel als uA741, MC1741, SN72741, TBA221 und ähnliche. Leider steht uns beim C64 direkt keine negative Spannung zur Verfügung, wie sie eigentlich beim Betrieb eines Operationsverstärkers nötig ist. Wir greifen daher in diesem Fall etwas in die Trickkiste und gaukeln dem IC eine symmetrische Spannungsversorgung vor, indem der positive Eingang mit Hilfe eines Spannungsteilers auf ein künstliches Massepotential (einen sogenannten virtuellen Nullpunkt) festgelegt ist. Die Betriebsspannungsanschlüsse führen einerseits 0 Volt (Masse) und andererseits die mit einer Kaskade erzeugte, und mittels eines Spannungsreglers auf 24 Volt stabilisierte positive Speisespannung. Lesen Sie hierzu bitte die eingehenden Beschreibungen in Kapitel 7.

Bild 2.36, Seite 84, zeigt die gesamte Wandlerschaltung. Betrachten wir zunächst einmal das Zusammenspiel zwischen Eingängen und Ausgang. Tritt am invertierenden Eingang (-) gegenüber dem nicht invertierenden (+) eine positive Spannung auf, dann erscheint sie sofort hoch verstärkt in entgegengesetzter Polarität am Ausgang und wird gleichzeitig über den Gegenkopplungswiderstand R_G auf den invertierenden Eingang zurückgeführt. Durch diesen Regelkreis stellt sich sehr schnell wieder ein Zustand ein, bei dem die Eingangsspannungsdifferenz 0 Volt beträgt.

Der Eingangswiderstand eines Operationsverstärkers ist sehr groß, und man kann davon ausgehen, daß kein Strom in ihn hineinfließt. Damit wird die Ausgangsspannung in dieser Schaltung allein bestimmt durch den Gegenkopplungs-

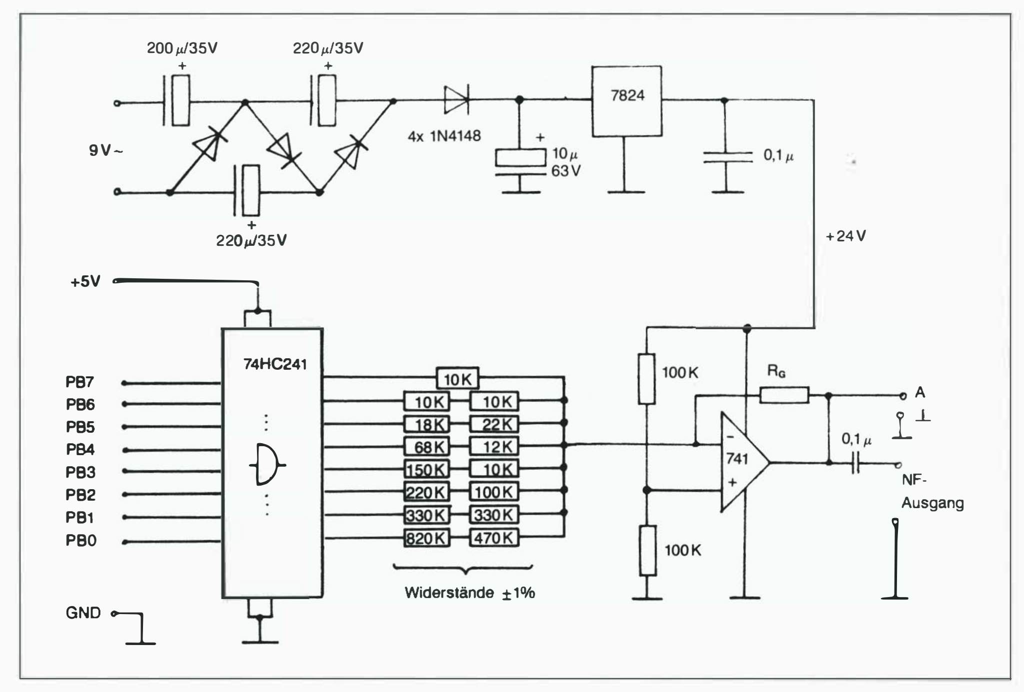


Bild 2.36: Schaltplan des D/A-Wandlers für NF-Zwecke

widerstand R_G und durch die Summe der Ströme in den acht Eingangszweigen. Mit unterschiedlicher Dimensionierung der Widerstände können die Eingangsspannungen verschieden hoch verstärkt werden. Trotz der unterschiedlichen Verhältnisse stellt sich jedoch immer ein virtueller Nullpunkt für alle Eingangsströme ein, so daß diese sich nicht gegenseitig beeinflussen können.

Rechnerisch ergibt sich für die Ausgangsspannung:

$$UA = -\left(\frac{R_G}{R_{EO}} \cdot U_{EO} + \frac{R_G}{R_{E1}} \cdot U_{E1} + \frac{R_G}{R_{E2}} \cdot U_{E2} + \ldots + \frac{R_G}{R_{E7}} \cdot U_{E7}\right)$$

Man muß nun dafür sorgen, daß jeder Eingang eine seiner Wertigkeit entsprechende Spannung hervorruft. Das wird jeweils durch einen genau passenden Eingangswiderstand erreicht. Es sollten handelsübliche und leicht zu beschaffende Normwiderstände mit 1% Toleranz verwendet werden. Daher sind fast alle Zweige aus zwei Einzelwiderständen zusammengesetzt. Tabelle 2.10 gibt einen Überblick.

Es sind zwei Amplitudenwerte für zwei verschiedene Gegenkopplungswiderstände angegeben. Bei diesen Spannungswerten handelt es sich aber jeweils um Abweichungen von der Normalspannung am Ausgang. Wegen des virtuellen Massepunktes führt der Ausgang nämlich einen recht hohen Gleichanteil von etwa 12 Volt. Die angegebene Ausgangsspannung muß

Eingang	Widerstand	realisiert durch	Spng. $R_G = 1K$	$R_G = 10K$
0	1,28M	820K + 470K	3,8 mV	0,038 V
1	640K	330K + 330K	7,5 mV	0,075 V
2	320K	220K + 100K	15,0 mV	0,150 V
3	160K	150K + 10K	31,25mV	0,312 V
4	80K	68K + 12K	62,5 mV	0,625 V
5	40K	18K + 22K	125,0 mV	1,250 V
6	20K	10K + 10K	250,0 mV	2,500 V
7	10K	10K	500,0 mV	5,000 V
maximal	(alle Eingänge HIGH)		995,35mV	9,953 V

Tabelle 2.10: Dimensionierung des D/A-Wandlers

von dem Gleichanteil subtrahiert werden.
Unsere Platine ist aber hauptsächlich für NF-Zwecke bestimmt, bei denen nur der Wechselspannungsanteil relevant ist. Dieser wird vor der DIN-Buchse mittels eines Kondensators abgetrennt. Für spezielle Anwendungen ist es jedoch auch möglich, die Ausgangsspannung am Punkt A direkt abzugreifen. Dazu wurde ein Lötstiftpaar auf der Platine vorgesehen.

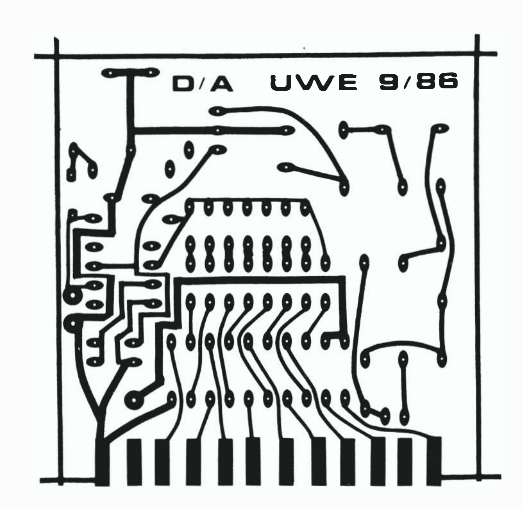


Bild 2.37: Layout des D/A-Wandlers

2.2.11.2 Aufbau der D/A-Wandlerplatine

Aufgrund der einfachen Schaltung mit gängigen Bauteilen ist die Wandlerkarte sehr preisgünstig herstellbar. Das Layout zeigt Bild 2.37. In Tabelle 2.11 sind die benötigten Bauteile zusammengestellt. Richten Sie sich nach den ausführlichen Hinweisen im Anhang A und dem Bestückungsplan in Bild 2.38, Seite 86.
Alle Widerstände außer R_G müssen aus Platzgründen stehend eingelötet werden. Der Platinendirektstecker wird auch hier mit seiner unteren Kontaktleiste direkt auf die Platine gelötet. Vergessen Sie nicht die drei Lötbrücken zur oberen Kontaktleiste. Der Spannungsregler muß mit der Metallplatte am Gehäuse zur Buchse hin eingelötet werden.
Foto 2.4, Seite 86, zeigt die Platine vor dem Einbau der Widerstände.
Über die DIN-Buchse kann die fertig bestückte Platine an jeden beliebigen Verstärker angeschlossen werden. Auch der Betrieb an der Verstärkerplatine aus Kapitel 5.3 ist möglich.
Zur Eindämmung von Störgeräuschen sollte als Verbindungskabel eine abge-

1 =	10 KΩ
2 =	22 KΩ
3 =	12 KΩ
4 =	10 KΩ
5 =	100 KΩ
6 =	330 KΩ
7 =	470 KΩ
8 =	10 KΩ
9 =	10 KΩ
10 =	18 KΩ
11 =	68 KΩ
12 =	150 KΩ
13 =	220 KΩ
14 =	330 KΩ
15 =	820 KΩ

Bild 2.38: Bestückungsplan des D/A-Wandlers

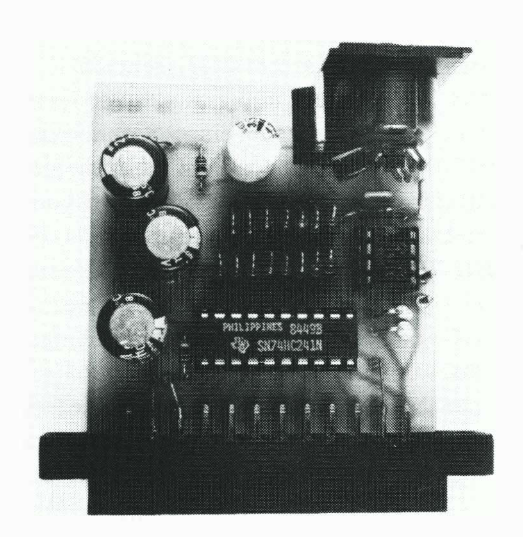


Foto 2.4: Der D/A-Wandler

schirmte Ausführung gewählt werden. Dabei muß das äußere Metallgeflecht an Masse liegen.

2.2.11.3 D/A-Wandlung für größere Ausgangsströme

Die gerade vorgestellte Schaltung hat einen Nachteil. Ihr lassen sich nämlich kaum größere Ströme entlocken, wie sie etwa zur Ansteuerung von Gleichstrommotoren nötig sind.

Dieses Problem kann jedoch mit der Schaltung nach Bild 2.39 gelöst werden. Beim L200 handelt es sich um einen positiven 5-Pin-Spannungsregler in einem Pentawatt-Gehäuse. Gegenüber den bekannten 3-Pin-Spannungsreglern (siehe Kapitel 7) sind beim L200 zwei weitere Anschlüsse für die Spannungs- und Stromprogrammierung vorhanden. Neben dem Schutz gegen kurzzeitige Eingangsspannungsspitzen bis zu 60 Volt besitzt auch der L200 weitere Maßnahmen

1	Achtfach-Treiber 74HC241 (auch LS möglich)	1	Widerstand 100 Kiloohm, ± 1%
1	Operationsverstärker 741	1	Widerstand 68 Kiloohm, ± 1%
1	IC-Sockel 20pol	1	Widerstand 22 Kiloohm, ± 1%
1	IC-Sockel 8pol	1	Widerstand 18 Kiloohm, ± 1%
1	Spannungsregler 7824	1	Widerstand 12 Kiloohm, ± 1%
4	Dioden 1N4148	4	Widerstände 10 Kiloohm, ± 1%
2	Widerstände 100 Kiloohm	3	Elektrolytkondensatoren 220 Mikrofarad / 35 Volt, radial
1	Widerstand 1 Kiloohm (RG, siehe Text)	1	Elektrolytkondensator 10 Mikrofarad / 63 Volt, radial
1	Widerstand 820 Kiloohm, ± 1%	2	Kondensatoren 0,1 Mikrofarad
1	Widerstand 470 Kiloohm, ± 1%	1	DIN-Buchse, 5pol, direkt einlötbar
2	Widerstände 330 Kiloohm, ± 1%	2	Lötstifte
1	Widerstand 220 Kiloohm, ± 1%	1	einseitige Platine nach Bild 2.37
1	Widerstand 150 Kiloohm, ± 1%		

Tabelle 2.11: Die Bauteile zum D/A-Wandler

gegen ausgangsseitigen Kurzschluß und thermische Überlastung.

Bild 2.39 zeigt eine Schaltung zur digitalen Programmierung der Ausgangsspannung. Für die Dimensionierung gilt die Formel

$$U_{aus} = 2{,}75V * (1 + \frac{R2}{R1})$$

Es ist lediglich darauf zu achten, daß die Sättigungsspannung der Schalttransistoren niedrig ist. Empfehlenswert ist der Typ BC378.

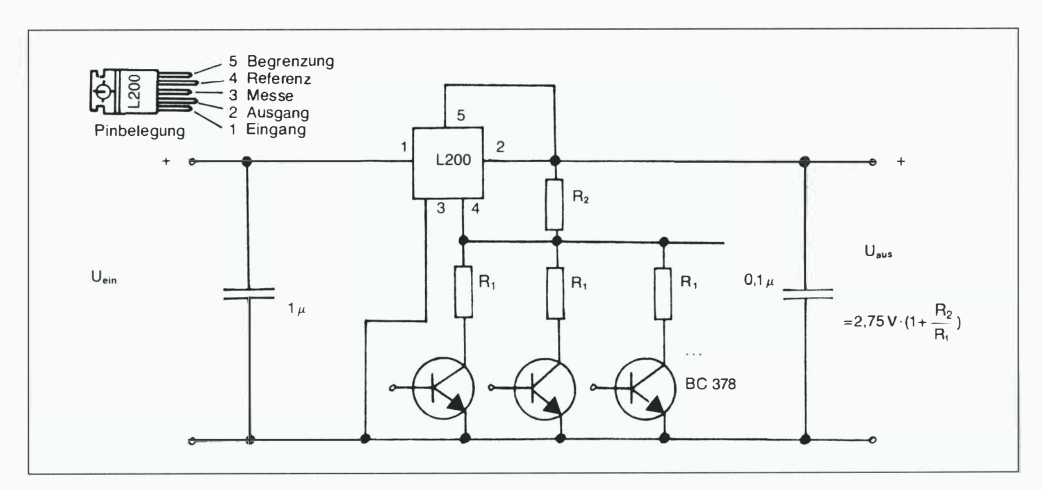


Bild 2.39: Digitale Spannungseinstellung mit dem L200

2.3 Die Handshake-Leitungen des CIA 6526

2.3.1 Hardwaregegebenheiten

Wie wir schon wissen, unterstützen auch die beiden CIA-Bausteine den Handshake-Datenaustausch. Das geschieht auf sehr einfache Weise:
Schauen Sie sich bitte in der CIA-Tabelle (Anhang B) das Interruptkontrollregister (Nummer 13) genauer an. Bit 4 wird immer gesetzt, wenn am Anschluß $\overline{\text{FLAG}}$ des entsprechenden Bausteins ein HIGH-LOW-Übergang aufgetreten ist. Damit steht uns ein leistungsfähiger Eingang zur Verfügung. Ein dort eintreffender Impuls – so kurz er auch sein mag – setzt in jedem Fall über ein internes Flipflop das erwähnte Bit 4. Der Computer ist dadurch nicht darauf angewiesen, ständig in möglichst kurzen Zeitabständen einen Eingang abzufragen, sondern kann etwas ganz anderes tun und braucht nur gelegentlich nachzuschauen, ob das $\overline{\text{FLAG}}$-Bit schon gesetzt ist. Damit werden externe Abläufe sehr einfach überwachbar. Denkbar wäre zum Beispiel die Datenausgabe auf einen Drucker innerhalb der Interruptschleife, während man mit dem Computer im Hauptprogramm beispielsweise bereits einen anderen Text editiert.
Zu beachten ist jedoch, daß jeder Lesezugriff auf das Interruptkontrollregister – zum Beispiel PRINT PEEK(56589) – alle darin befindlichen Informationsbits automatisch löscht. Dieser Effekt ist im allgemeinen sehr nützlich und spart bei richtigem Einsatz Programmieraufwand und Zeit. Zum Löschen des Registers vor der Benutzung braucht man zum Beispiel lediglich eine Leseoperation auszuführen.
Die Meldeleitung des Computers ist der CIA-Anschluß $\overline{\text{PC}}$. Er ist fest verdrahtet, also nicht programmierbar, und gibt nach jedem Zugriff auf das Portregister einen LOW-Impuls mit der Länge eines Systemtaktes aus. Sobald man also zum Beispiel mit POKE 56577,61 einen Wert in den User-Port schreibt, wird $\overline{\text{PC2}}$ kurz LOW und dann wieder HIGH.
Gleiches passiert bei einem Lesezugriff, zum Beispiel PRINT PEEK(56577).
Achtung! Der Anschluß $\overline{\text{PC}}$ ist ein Ausgang mit offenem Kollektor. Das heißt, daß zum Betrieb unbedingt noch ein Pull-up-Widerstand nach + geschaltet werden muß. Ein Blick in den Schaltplan des C64 zeigt, daß an allen Steueranschlüssen solche Widerstände vorhanden sind, mit Ausnahme von $\overline{\text{PC}}$. Diese Tatsache hat ihren Grund darin, daß man den Pull-up-Widerstand ans andere Ende der Übertragungsstrecke setzen kann. So ist die Leitung relativ niederohmig abgeschlossen und wird störungsunempfindlicher. Außerdem ermöglicht der offene Kollektor auch Betrieb mit höheren Ausgangsspannungen.
Die Steueranschlüsse der beiden CIAs werden am User-Port mit den Zusätzen 1 und 2 unterschieden. $\overline{\text{FLAG2}}$ ist also der Handshake-Eingang und $\overline{\text{PC2}}$ der Handshake-Ausgang des CIA # 2, der für den User-Port verantwortlich zeichnet. Neben diesen Pins sind am User-Port noch die Anschlüsse CNT und SP beider CIAs herausgeführt, auf die wir später zu sprechen kommen werden.

2.3.2 Der User-Port als Centronics-Schnittstelle

2.3.2.1 Die ASCII-Norm

Wie wir beim Windrichtungs-Anzeiger schon gesehen haben, lassen sich bei gleichzeitiger Benutzung mehrerer Leitungen viele verschiedene Kombinationen erzeugen. Jede Ader liefert 1 Bit. Mit einem einzigen Bit kann man nur zwei verschiedene Zustände erreichen (HIGH oder LOW). 2 Bit ermöglichen bereits vier Kombinationen, 3 Bit acht, 4 Bit 16 und so weiter. Mit den acht am User-Port zur Verfügung stehenden Leitungen kann man demnach 256 verschiedene Kombinationen erreichen.

Um sinnvoll Informationen übertragen zu können, die nicht nur aus Zahlen bestehen, muß natürlich vorher festgelegt werden, was jede Bitkombination bedeuten soll. Die Zuordnung ist grundsätzlich von Fall zu Fall frei wählbar, damit man aber Geräte verschiedener Hersteller möglichst problemlos miteinander verbinden kann, haben sich mehrere Firmen in Amerika schon früh auf eine gemeinsame Darstellungsweise geeinigt. Es entstand der »American Standard Code for Information Interchange« (Amerikanischer Standard-Code für Informationsaustausch) ASCII, der heute weltweit einen Standard bildet.

Anhang E enthält eine Tabelle mit allen 256 Kombinationsmöglichkeiten. Die Hälfte davon ist durch den ASCII-Code fest belegt. Er arbeitet also mit nur 7 Bit, um das achte für Sonderfunktionen (zum Beispiel Parität) freizuhalten. Aber auch die dann verbleibenden 128 darstellbaren Zeichen enthalten alle üblichen Groß- und Kleinbuchstaben, Ziffern und Sonderzeichen. Die ersten 31 Codes bilden Steuerzeichen für Drucker und Terminals. Sie werden ebenfalls im Anhang E genauer erläutert.

2.3.2.2 Datenaustausch mit Händeschütteln

Zwei Geräte, die man miteinander verbinden will, beispielsweise ein Computer und ein Drucker, werden im Normalfall nicht exakt die gleiche Arbeitsgeschwindigkeit haben. Es genügt also nicht, eine Folge von ASCII-Bitkombinationen nacheinander auf die Verbindungsleitungen zu legen, und sich darauf zu verlassen, daß sie am anderen Ende schon richtig erkannt werden. Vielmehr müssen die beiden Geräte Hand in Hand arbeiten. Dazu stellt man jeweils eine Meldeleitung bereit. Die Übertragung zwischen zwei Geräten könnte damit folgendermaßen ablaufen:

1. Der Sender legt die Kombination für das zu übermittelnde Zeichen an die Datenleitungen und
2. gibt auf seiner Meldeleitung einen kurzen Impuls aus, um zu signalisieren, daß neue Daten anliegen.
3. Der Empfänger holt sich daraufhin die Daten ab, verarbeitet sie und
4. teilt seine Empfangsbereitschaft durch einen kurzen Impuls auf einer anderen Meldeleitung dem Partner mit.

Danach fängt der nächste Zyklus wieder bei Punkt 1 an.

Dieses Verfahren nennt man »Handshake«-Übertragung (Handshake = Händeschütteln). Die beiden Geräte können intern mit völlig verschiedenen Geschwindigkeiten arbeiten, ohne daß Probleme auftreten, da

die Abtastzeitpunkte durch die Handshake-Signale genau definiert sind.

Der Druckerhersteller Centronics führte eine Schnittstelle ein, die bald von zahlreichen anderen Firmen übernommen wurde und heute einen Quasi-Standard für Drucker darstellt. Außer dem gerade vorgestellten Übertragungsprotokoll ist hier noch eine andere Synchronisationsmöglichkeit zwischen den Geräten vorgesehen. Verantwortlich dafür ist die Leitung BUSY, die vom Drucker bedient wird (engl. busy = beschäftigt). Der erste Teil der Übertragung läuft genau gleich ab:

1. Der Computer legt die Kombination für das zu druckende Zeichen an die Datenleitungen und
2. gibt auf seiner Meldeleitung ($\overline{\text{STROBE}}$) einen kurzen Impuls aus, um zu signalisieren, daß die Daten anliegen.
3. Der Drucker setzt daraufhin zunächst die Leitung BUSY auf HIGH, liest die Kombination ein, druckt das Zeichen und
4. setzt dann erst die Leitung BUSY zurück auf LOW, um dem Computer seine Empfangsbereitschaft anzuzeigen.

Tabelle 2.12 zeigt die genormte Belegung des verwendeten 36poligen Steckers, der in Bild 2.40 zu sehen ist. Es ist nicht erforderlich, alle Schnittstellensignale zu erzeugen bzw. auszuwerten. Für einen einwandfreien Betrieb genügen bereits die 8 Datenbit sowie die Signale BUSY und $\overline{\text{STROBE}}$. Alle anderen Pins können unbeschaltet bleiben. Sie sind teilweise leider auch von Hersteller zu Hersteller unterschiedlich belegt.

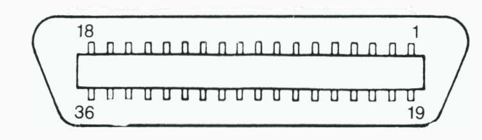


Bild 2.40: Außenansicht Centronics-Normbuchse bzw. Stecker mit Sicht auf Lötseite

Der Anschluß $\overline{\text{INIT}}$ (Signalpin 31) ist der Reset-Eingang des Druckers. Mit einem LOW-Impuls kann er hier von Hand auf die vorgewählten Werte zurückgesetzt wer-

Signalpin	Massepin	Signal	Richtung	Bedeutung
1	19	$\overline{\text{STROBE}}$	Eingang	Bei der fallenden Flanke von $\overline{\text{STROBE}}$ werden die Daten vom Drucker übernommen. $>0{,}5\mu s$
2	20	DATA 1	Eingang	Datenbit
3	21	DATA 2	Eingang	Datenbit
4	22	DATA 3	Eingang	Datenbit
5	23	DATA 4	Eingang	Datenbit
6	24	DATA 5	Eingang	Datenbit
7	25	DATA 6	Eingang	Datenbit
8	26	DATA 7	Eingang	Datenbit

Signalpin	Massepin	Signal	Richtung	Bedeutung
9	27	DATA 8	Eingang	Datenbit
10	28	$\overline{\text{ACKNLG}}$	Ausgang	LOW-Impuls von ca. 10μs Länge. Besagt, daß die Daten verarbeitet wurden und der Drucker wieder bereit ist.
11	29	BUSY	Ausgang	Solange BUSY HIGH-Signal führt, ist der Drucker nicht empfangsbereit.
12	30	PE	Ausgang	HIGH = Papierende
13	-	SELECTED	Ausgang	Geht nach LOW, wenn der Drucker OFF-LINE ist
14	-	$\overline{\text{AUTOFEEDXT}}$	Eingang	Linefeed-Steuerung LOW = automatisch HIGH = per Befehl
15	-	NC	-	unbenutzt
16	-	0V	-	Massepegel (Logik)
17	-	CHASSIS GND	-	Masse Drucker (isoliert von Logikmasse)
18	-	+5V	-	Versorgungsspannung
19-30	-	GND	-	Massepins für 1-12
31	-	$\overline{\text{INIT}}$	Eingang	Druckerinitialisierung bei LOW-Impuls länger 50μs
32	-	$\overline{\text{ERROR}}$	Ausgang	Geht auf LOW, wenn 1. Drucker offline, 2. Papierende, 3. Fehler entdeckt.
33	-	GND	-	Massepins wie 19-30
34	-	NC	-	unbenutzt
35	-	-	-	HIGH-Pegel
36	-	$\overline{\text{SLCTIN}}$	Eingang	Druckerselektion Codes DC1/DC3 funktionieren nur, wenn dieser Pin HIGH ist.

Tabelle 2.12: Belegung des Centronics-Steckers eines Druckers

den. Sie können dazu die Tasterschaltung nach Bild 2.15 verwenden, die unter Umständen sogar noch ins Steckergehäuse paßt.

Pin 18 führt bei manchen Druckern (zum Beispiel STAR) die +5-V-Versorgungsspannung der internen Elektronik. Das ist sehr nützlich für kleine Interface-Schaltungen, die so – ohne zusätzliche Stromversorgung – einfach in die Zuleitung gebaut werden können.

2.3.2.3 Schon alles da!

Aus dem User-Port des C64 läßt sich – allein mit etwas Software – leicht eine Centronics-Schnittstelle zaubern. Dazu können die eben beschriebenen Handshake-Leitungen des CIA-Bausteins 6526 benutzt werden. Üblicherweise verwendet man jedoch nur den Handshake-Eingang $\overline{\text{FLAG2}}$ zur Auswertung des BUSY-Signals und benutzt den freien Anschluß von Port A (PA2) zur Ausgabe des $\overline{\text{STROBE}}$-Impulses.

Inzwischen existieren schon genug Programme, die den User-Port am C64 oder C128 zu einer vollwertigen Centronics-Schnittstelle machen. Ein guter Grund, bei der Kaufentscheidung für einen Drucker nicht nur auf Commodore-Kompatibilität zu schauen. Es empfiehlt sich sogar, einen Centronics-Drucker zu kaufen, denn man kann schließlich nie wissen, ob man nicht doch eines fernen Tages auf einen anderen Rechner umsteigt. Nach einem seriellen Bus wird man dort meist vergebens suchen.

Der Anschluß eines universellen Druckers ist einfach:

Man nehme einen User-Port-Stecker auf der einen Seite, den beschriebenen 36poligen Stecker auf der anderen, dazwischen ein mindestens 11poliges Kabel mit höchstens 1,5m Länge, am besten mit Abschirmgeflecht. Das Ganze wird nach Bild 2.41 verbunden – fertig.

Doch bevor Sie sich an die Arbeit machen, lesen Sie lieber noch ein Stückchen weiter. Der Teufel steckt nämlich im Detail und Ihr CIA-Baustein steht auf dem Spiel!

2.3.2.4 CIA in Lebensgefahr

Ein langes Kabel ist eben keine ideale elektrische Verbindung, wie uns der schlichte

Userport			Druckerstecker	
PA2	M		1	$\overline{\text{STROBE}}$
PB0	C		2	DATA 1
PB1	D		3	DATA 2
PB2	E		4	DATA 3
PB3	F		5	DATA 4
PB4	H		6	DATA 5
PB5	J		7	DATA 6
PB6	K		8	DATA 7
PB7	L		9	DATA 8
$\overline{\text{FLAG2}}$	B		10	$\overline{\text{ACK}}$
GND	A		19	Logik GND
		Schirmung	17	Chassis GND

Bild 2.41: Schaltung des Centronics-Kabels für Längen bis 1,5 m

Strich im Schaltplan glauben machen will. Da gibt es nicht nur den allgemein bekannten spezifischen Widerstand im Leiter, sondern auch kleine Induktivitäten, wenn das Kabel zum Beispiel in einer Schlaufe liegt, und kapazitive Kopplungen der dicht nebeneinander geführten Adern untereinander. Alle diese unerwünschten Effekte wachsen mit zunehmender Kabellänge. Bild 2.42 zeigt eine Anordnung, die den Eigenschaften eines langen Kabels schon eher entspricht; und hier ist nur ein einziges Leiterpaar gezeichnet!
Nehmen wir nur einmal die kapazitive Wechselwirkung zwischen einer Daten- und der Masseleitung heraus. Zunächst sollen beide gleiches elektrisches Potential besitzen, die Datenleitung soll also auf Masse liegen. Die beiden betrachteten Adern sind auf der gesamten Kabellänge dicht nebeneinander geführt, nur durch ihre Isolierungen getrennt. Sie bilden damit einen Kondensator. Jetzt wechselt die Datenleitung innerhalb sehr kurzer Zeit von 0 Volt auf +5 Volt. Der Kondensator muß sich sehr schnell aufladen und entnimmt dabei dem treibenden CIA im ersten Moment einen sehr großen Strom. Genau dasselbe passiert auch wieder beim Umladen in den Ausgangszustand. Je größer die Kabellänge und damit der Wert des gebildeten Kondensators ist, um so größer wird die Strombelastung für den CIA-Baustein. Irgendwann einmal brennt einer seiner Leitungstreiber durch und macht ihn damit – zumindest was den User-Port angeht – unbrauchbar.

2.3.2.5 Treiber als Vorposten

Sie brauchen aber nun nicht gleich zu verzweifeln oder den Drucker näher heranzurücken und das lange Kabel abzuschneiden, denn mit externen Bausteinen kann man ja dem CIA etwas Arbeit abnehmen. Bild 2.43 zeigt eine geeignete Schaltung. Bei dem 7407 handelt es sich um einen speziellen Treiberbaustein mit offenem Kollektor, der wesentlich mehr Strom liefern kann als der CIA. Für jede Ader sind bis zu 40 mA zulässig. Steuert man diesen Treiber über ein kurzes Kabel mit dem Computer an, dann darf der Drucker auch ruhig mal fünf Meter weiter stehen. Und sollte wirklich einmal etwas schiefgehen, dann sind allerhöchstens die beiden Treiberbausteine hin, und Sie sind mit fünf Mark wieder dabei. Die bei Open-

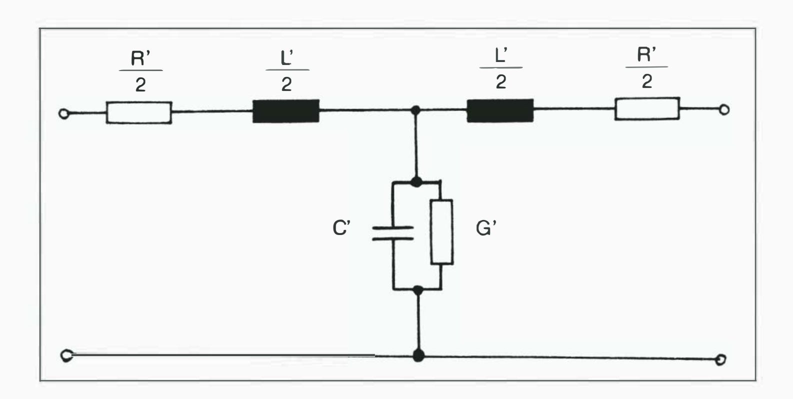


Bild 2.42: Ersatzschaltung eines Leitungselements

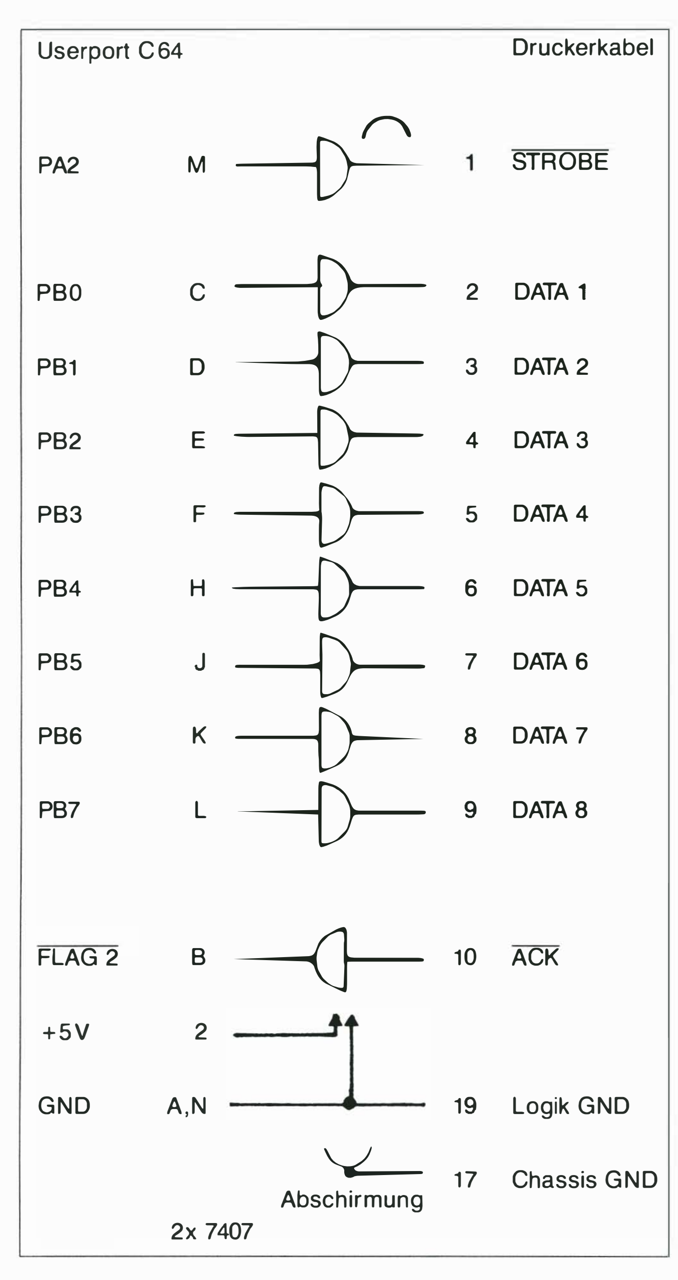


Bild 2.43: Schaltplan des Drucker-Treibers

Collector-Schaltungen zur Funktion nötigen Pull-up-Widerstände sind standardmäßig bereits im Drucker enthalten. Wie bereits gesagt, ist es bei langen Übertragungsleitungen vorteilhaft, diese Widerstände erst auf der Empfängerseite einzufügen.

Eine Platine nach dem abgebildeten Lay-

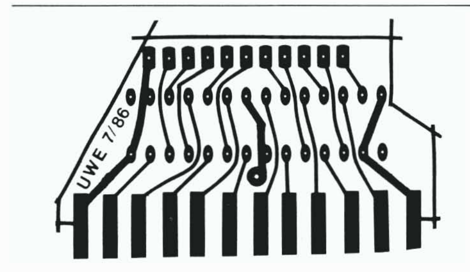


Bild 2.44: Layout für den Drucker-Treiber

out (Bild 2.44) kann direkt am User-Port-Stecker angelötet werden und verschwindet dann mit ihm in einem üblichen Steckergehäuse. Die wenigen Bauteile – siehe in Tabelle 2.13 – werden nach dem Bestückungsplan (Bild 2.45) auf der vorher exakt ausgeschnittenen und in das Steckergehäuse eingepaßten Platine verlötet. Beachten Sie die Hinweise zur Platinenherstellung und Bestückung in Anhang A.

Der User-Port-Stecker wird mit seiner unteren Kontaktleiste auf der Platine verlötet. Vergessen Sie nicht die Verbindung für +5-V-Versorgungsspannung. Es ist der einzige benutzte Kontakt der oberen Leiste am User-Port-Stecker und befindet sich vom Computer aus gesehen an zweiter Stelle von rechts (Anschluß 2). Damit die Platine später in ein übliches Steckergehäuse paßt, dürfen die ICs hier ausnahmsweise nicht gesockelt werden. Bei der Bauteilbeschaffung ist unbedingt darauf zu achten, ICs mit kleinen Gehäusen zu

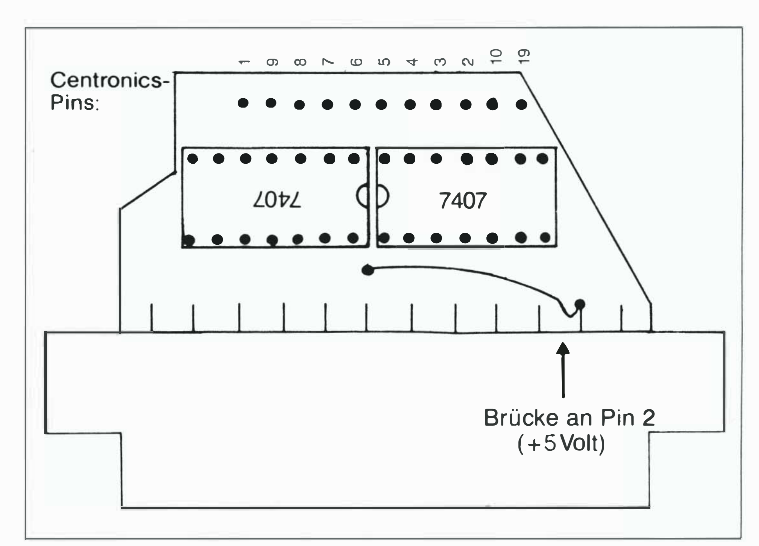


Bild 2.45: Bestückungsplan des Drucker-Treibers

2	Treiber 7407 (anreihbare Gehäuseform)
1	User-Port-Stecker (z.B. Cardcon-Stecker Nr. 251-12-50-170 von TRW)
1	User-Port-Steckergehäuse
1	einseitige Platine nach Bild 2.44
1	Kabel 11pol, abgeschirmt
1	Centronics-Stecker (36polig Amphenol, z.B. AMP 57-30360)

Tabelle 2.13: Die Einzelteile zum Drucker-Treiber

bekommen. Sie müssen aus Platzgründen direkt im 2,54-mm-Raster anreihbar sein.
Das verwendete Kabel muß auch hier mindestens 11polig sein und sollte eine Abschirmung besitzen. Für optimale Wirkung darf diese Abschirmung nur am Centronics-Stecker an Masse gelegt werden und muß am anderen frei bleiben. Nur so bleibt sie stromlos, und es ist sichergestellt, daß an ihr kein Potentialabfall entlang des Kabels auftreten kann. Ebenso sollten eventuell überzählige Adern an diesem Ende mit an Masse gelegt werden, während sie auf der anderen Seite unbeschaltet bleiben. Diese Maßnahme verstärkt den Abschirmeffekt.

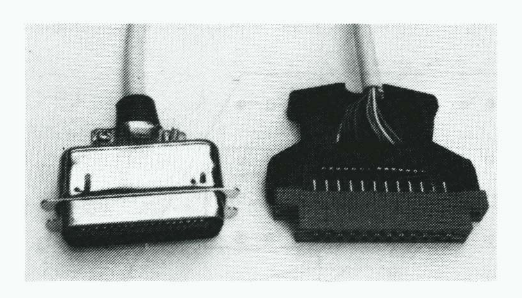


Foto 2.5: Das Centronics-Kabel mit Treiber

Foto 2.5 zeigt die fertige Platine mit angelötetem Kabel vor dem Einbau ins User-Port-Steckergehäuse. Auf beiden Seiten muß das Kabel zur Zugentlastung fest angeschraubt werden, damit auch bei grober Behandlung keine Aderunterbrechungen oder gar Kurzschlüsse entstehen.
Der Betrieb ist nun wie bei allen User-Port-Centronics-Kabeln möglich: entweder von Programmen aus, die diese Schnittstelle von sich aus mitunterstützen, durch Nachladen eines geeigneten Softwaretreibers, oder mittels geänderter Druckerroutinen im Kernal-ROM des Computers.

2.3.2.6 Die Centronics-Software

Leider verwendet Commodore intern nicht die ASCII-Norm zur Darstellung von Zeichen. Daher müssen im Druckprogramm die entsprechenden Codes erst nach ASCII gewandelt werden. Ein weiteres Problem stellen die Commodore-Sonderzeichen dar, die normalerweise bei Universaldruckern nicht im Zeichensatz vorhanden sind.
Die beiliegende Diskette enthält eine Super-Centronics-Schnittstelle am User-Port, die in ihren Leistungen weit über vergleichbare Hilfsroutinen hinausgeht, da sie verschiedene Betriebsmodi besitzt. Der Assembler-Quelltext heißt CENTRONICS.ASM, der assemblierte Objectcode CENTRONICS.OBJ.
Für normalen Einsatz laden Sie bitte den BASIC-Lader CENTRONICS.BAS und starten ihn mit RUN. Der C64 installiert daraufhin die Hilfsroutine im Bereich $CC00 bis $CDC7, sorgt dafür, daß sie bei einem Druckbefehl aufgerufen wird, und meldet sich anschließend mit READY, als wäre nichts gewesen.
Nun kann man wie gewöhnlich mit OPEN 1,4 einen Ausgabekanal zum Drucker öffnen und mit PRINT#1"Text" Zeichen ausgeben.
Eine andere Möglichkeit bietet sich mit dem Befehl OPEN 1,5. Der Ausdruck erfolgt dann nicht mehr mit Groß/Grafik-Zeichen, sondern in Groß/Kleinschrift, wie nach dem Umschalten des Zeichensatzes im C64 durch SHIFT/Commodore.
Durch die Umleitung der Bildschirmausgabe auf einen geöffneten Kanal mit dem BASIC-Befehl CMD werden alle Zeichen genau wie gewohnt statt auf dem Bild-

schirm auf den Drucker ausgegeben. Da BASIC-Schlüsselworte zwar als Tokens gespeichert werden, beim Anzeigen aber wieder als Worte erscheinen, nimmt der Rechner intern eine Wandlung vor. Tokens erscheinen also ebenfalls auf dem Drucker als Zeichenketten. Dieser Effekt wirkt sich aber nicht störend aus, da Tokens keine ASCII-Zeichen sind und auch im Drucker keine sinnvolle Reaktion auslösen würden. Im Gegenteil ist eine Wandlung bei den meisten Anwendungen sogar sinnvoll. Beispielsweise speichert auch ProfiAss alle in den Kommentarzeilen des Quelltextes enthaltenen Schlüsselworte als Tokens ab, da er mit dem normalen BASIC-Editor arbeitet. Beim Ausdruck werden diese Codes dann erst wieder entzerrt.

Ein besonderer Leckerbissen des Centronics-Programms ist aber der Ausdruck mit OPEN 1,6. Hierbei arbeitet der Drucker im Grafikmodus und druckt die Zeichen genau so, wie sie auf dem Bildschirm erscheinen. Dabei ist es egal, ob der Original-Zeichensatz selektiert ist, oder ob eine andere Schriftart nachgeladen wurde. Das Programm erkennt automatisch, welcher Speicherbereich zuständig ist. In dieser Betriebsart sind endlich auch originalgetreue Listings möglich. Bisher bereiteten die reversen Steuerzeichen beim Ausdruck über einen Centronics-Drucker immer große Probleme. Diesem Manko ist hiermit ein Ende gesetzt.

2.3.3 Schnellere Diskettenübertragung

Jeden, der ein Diskettenlaufwerk besitzt, hat wohl schon einmal die lahme Übertragungsgeschwindigkeit der Floppydaten über den seriellen IEC-Bus geärgert. Eine parallele Übertragung dagegen kann grundsätzlich um sehr vieles schneller sein (siehe dazu nächstes Kapitel). Wie in Kapitel 2.2.9 gesagt, existiert in der Floppy noch ein freier 8-Bit-Port. Zusammen mit dem User-Port ist hier eine parallele Datenübertragung durchaus denkbar. Es kommen sogar immer mehr Programme auf den Markt, die eine solche Verbindung ausnutzen. Meist wird von einer Speed-Dos-kompatiblen Version gesprochen, da auch bei diesem Floppy-Speeder ein ähnliches Kabel Verwendung findet.

Auch die Datenübertragung mit der Floppy geschieht im Handshake-Betrieb, was sehr leicht zu realisieren ist, da der VIA-Baustein 6522 in der Floppy ebenfalls Handshake-Leitungen besitzt. Dort heißen sie CA und CB.

Auf die Programmierung soll jedoch hier nicht eingegangen werden.

Die Herstellung des Kabels geschieht nach Tabelle 2.15. Dabei ist es empfehlenswert – wie schon in Kapitel 2.2.9 vorgeschlagen – einen Zwischenstecker im Floppygehäuse anzubringen, um sich anderweitige Verwendungen offenzuhalten.

User-Port		VIA	6522
$\overline{\text{FLAG2}}$	B	39	CA2
PB0	C	2	PA0
PB1	D	3	PA1
PB2	E	4	PA2
PB3	F	5	PA3
PB4	H	6	PA4
PB5	J	7	PA5
PB6	K	8	PA6
PB7	L	9	PA7
$\overline{\text{PC2}}$	8	18	CB1

Tabelle 2.15: Anschluß eines Parallelkabels an User-Port und Floppy

2.3.4 Fernschalten per Telefon – gebührenfrei

Daß die Handshake-Leitungen auch bei ganz allgemeinen Schaltungen sehr gute Dienste leisten können, zeigen die folgenden Anwendungen. Als erste Schaltung folgt hier ein »Teleschalter«, mit dem Sie über bestimmte Klingelkombinationen mehrere Geräte ein- und ausschalten können.

2.3.4.1 Die Idee

Angenommen, Sie sind irgendwo in einem fernen Land und genießen Ihren wohlverdienten Urlaub. Natürlich möchten Sie nicht, daß Ihre wertvolle Computeranlage in der Zwischenzeit zu Hause geklaut wird. Wie wäre es, wenn Sie den Langfingern gerade mit Hilfe Ihres Computers vorspiegeln würden, Sie seien zu Hause? Kurz daheim anklingeln genügt, und schon schaltet sich beispielsweise die Schreibtischlampe und die Stereoanlage ein. Oder wie wäre es, wenn Sie kurz vor Feierabend per Telefon die Kaffeemaschine und die Heizung einschalten könnten, so daß Sie bei der Ankunft in Ihrer behaglich temperierten Wohnung schon vom köstlichen Duft frisch gebrühten Kaffees empfangen würden?
Es gibt unzählige Einsatzmöglichkeiten. Die Steuersoftware erlaubt es nämlich auch, eigene Maschinenroutinen durch Geräusche von außen aufzurufen.

2.3.4.2 Die Teleschalter-Hardware

Wie aus dem Schaltplan (Bild 2.46) zu erkennen ist, besteht der nötige Hardwarezusatz aus drei getrennten Teilen. Fangen wir einmal oben an. Sie erkennen sicherlich die Mikrofonschaltung aus Abschnitt 2.2.4.6 wieder. Bei der hier verwendeten Dimensionierung wird ein Optimum erreicht, wenn der Telefonapparat dicht neben dem Computer steht. Das Monoflop macht in der ebenfalls schon beschriebenen Weise aus den vielen kurzen, durch das Klingeln erzeugten Impulsen ein einziges Signal. Sein Ausgang ist mit dem Handshake-Eingang $\overline{\text{FLAG2}}$ des C 64 verbunden. Eine negative Schaltflanke an diesem Pin setzt im CIA-Baustein # 2 also das $\overline{\text{FLAG}}$-Bit und kann so abgefragt werden.
Nun zur zweiten Zeile des Schaltbildes, dem Autoreset-Teil. Der Telefonschalter soll unbeaufsichtigt auch über längere Zeiträume immer einsatzbereit sein. Es wäre sehr ärgerlich, wenn man von ferne etwas schalten wollte, während der Rechner aus irgendeinem Zufall schon längst abgestürzt ist. Um dem vorzubeugen, läßt man vom Programm periodisch einen Impuls am Anschluß $\overline{\text{PC2}}$ erzeugen. Dies geschieht durch einfaches Lesen der Portadresse $DD01 (= 56577 dezimal). Das zweite Monoflop im Baustein 74LS123 hat eine sehr lange Zeitkonstante von etwa 2,8 Sekunden. Kommt während dieser Zeit ein Impuls an seinen Eingang, so wird es erneut getriggert und sein Ausgang an Pin 4 bleibt auf LOW. Wird das Programm aber aus irgendeinem Grund unterbrochen, dann wechselt der Ausgang nach Ablauf der festgelegten Zeit auf HIGH und das nachgeschaltete Monoflop 74LS121 löst einen Reset am Computer aus. Hat nun das Programm einen Autostart, dann beginnt der Ablauf neu.
Falls dieser Autoreset nicht gewünscht wird, dann kann man den 74LS121 ein-

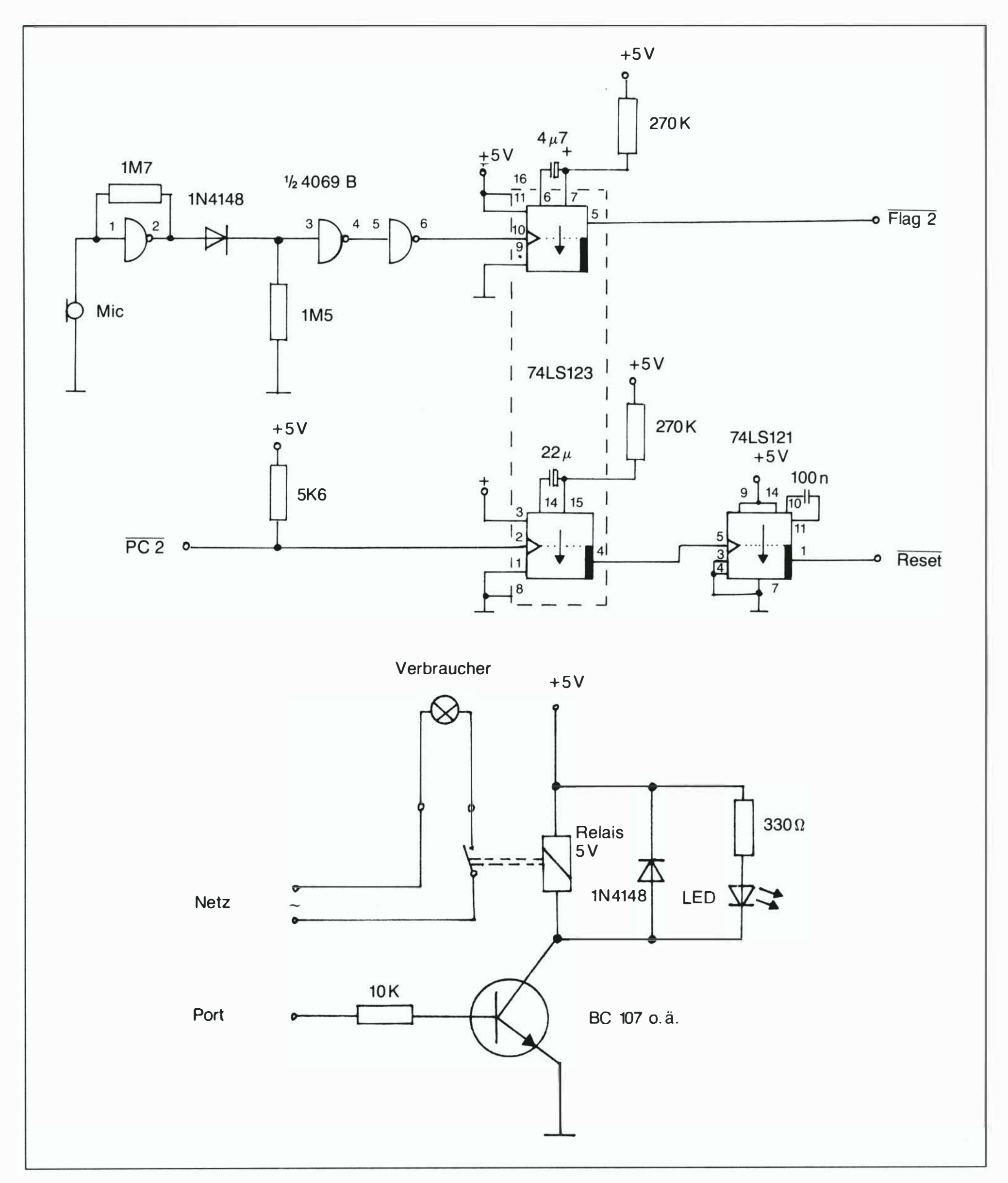


Bild 2.46: Schaltplan der Hardware-Erweiterung zum Teleschalter

fach aus seinem Sockel ziehen. Bei Stromausfälle hilft es natürlich nur, wenn man sein Steuerprogramm auf einem ROM-Modul untergebracht hat. Eine solche Platine werden wir in Kapitel 6.4 noch kennenlernen.

Die Laststufenschaltung im letzten Drittel des Schaltplans wurde ebenfalls bereits in

Kapitel 2.2.3 vorgestellt und muß für jeden benutzten Kanal getrennt aufgebaut werden. Auf der Platine sind acht Kanäle vorgesehen. Es ist jedoch nicht notwendig, sie alle voll zu bestücken. Man kann auch nur die Leuchtdiode mit Vorwiderstand im Kollektorzweig einlöten, und das Relais sowie die parallelgeschaltete Schutzdiode weglassen. So läßt sich jeweils der aktuelle Zustand der Ausgänge optisch überwachen. Die Diode 1N4148 sollte jedoch keinesfalls vergessen werden, wenn man das Relais einsetzt, denn sie schützt den Transistor vor gefährlichen Induktionsströmen, die entstehen können, wenn der Relaisanker beim Ausschalten des Kanals abfällt.

Außerdem wurde auf der Platine zum Schutz gegen Störungen noch ein Kondensator von 0,1 Mikrofarad zwischen dem positiven und negativen Pol der Speisespannung eingefügt.

2.3.4.3 Aufbau der Lauschplatine

Der Aufbau der Schaltung kann auf einer Platine nach dem abgebildeten Layout (Bild 2.47) anhand des Bestückungsplans (Bild 2.48) erfolgen. Tabelle 2.16, Seite 102, enthält die benötigten Einzelteile.

Sollen 220-V-Geräte über die Platine betrieben werden, muß unbedingt der gesamte Lastbereich, also alles, was mit Netzspannung in Berührung kommt, mit Isolierband abgeklebt werden.

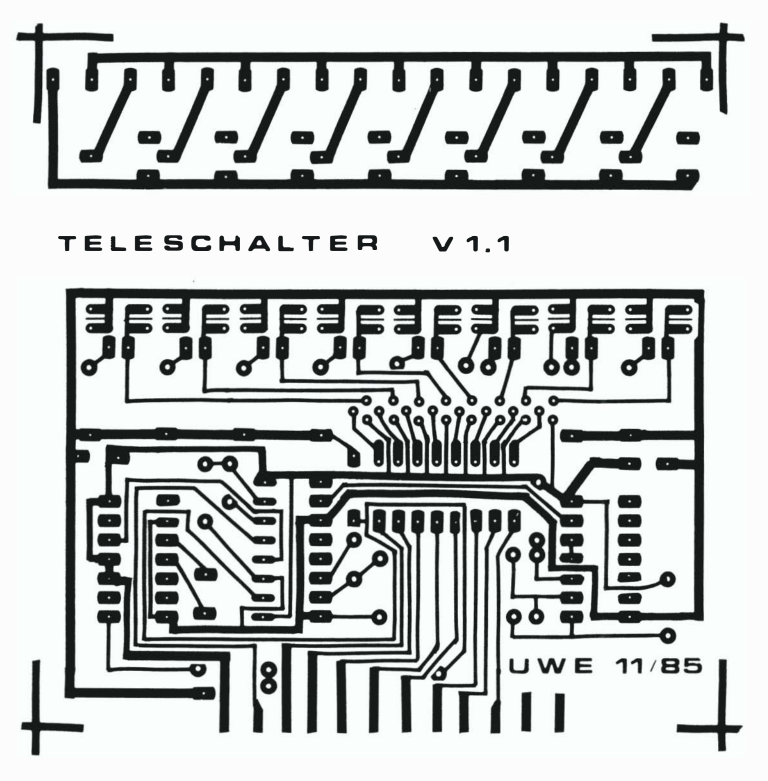


Bild 2.47: Das Layout zum Teleschalter

Wie in der Abbildung des Mustergerätes (Foto 2.6) zu sehen, wird der User-Port-Stecker mit seiner unteren Kontaktleiste direkt auf die Platine gelötet. Zur Oberseite werden lediglich drei Verbindungen benötigt, die mit Drahtbrücken, wie im Bestückungsplan angegeben, ausgeführt werden.

Nach dem Aufbau kann man mit dem kurzen Programm LAUFLICHT.DEMO die Port-Beschaltung überprüfen. Es läßt zyklisch jede LED aufleuchten und das entsprechende Relais anziehen – eine achtkanalige Lauflichtkette.

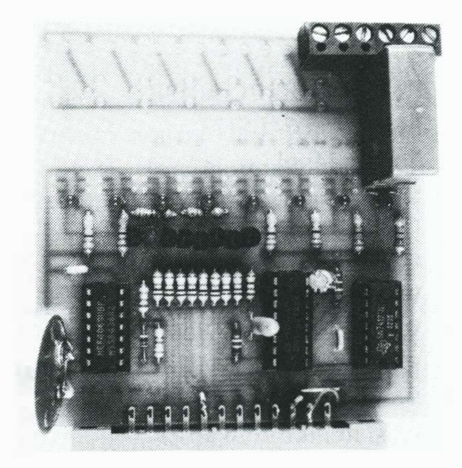


Foto 2.6: Der Teleschalter

Ausgangskanäle
zu schaltende Spannung ein
8 7 6 5 4 3 2 1
9 Lüsterklemmpaare
Relais 8 Relais 7 Relais 6 Relais 5 Relais 4 Relais 3 Relais 2 Relais 1
8 Relais
8x 1N4148
8x LED
330Ω
8x BC 107 o. ä.
0,1µ 1M5 5K6 4,7µ 270K 0,1µ
8x 10K
22µ
4069 74LS123 74121
1M7
270K
Mic.
1N4148
0,1µ
4 Brücken
Userport-Stecker
(auf Platinenunterseite)

Bild 2.48: Bestückung des Teleschalters

	Grundausbau:
1	CMOS-Logik-IC 4069B
1	Monoflop 74LS123
1	Sockel 14pol
1	Sockel 16pol
1	Diode 1N4148
1	Widerstand 1,7 Megaohm
1	Widerstand 1,5 Megaohm
1	Widerstand 270 Kiloohm
1	Kondensator 4,7 Mikrofarad
3	Kondensatoren 0,1 Mikrofarad
1	Kristallmikrofonkapsel
1	User-Port-Stecker
1	Lüsterklemmpaar, lötbar (z.B. Völkner Nr. 0307279)
1	einseitige Platine nach Bild 2.47
	Autoresetschaltung:
1	Monoflop 74LS121
1	Widerstand 270 Kiloohm
1	Widerstand 5,6 Kiloohm
1	Kondensator 22 Mikrofarad
1	Kondensator 100 Nanofarad
	je Schaltkanal:
1	Relais 1 x um (z.B. Conrad Nr. 491470)
1	Lüsterklemmpaar, lötbar (z.B. Völkner Nr. 0307279)
1	Schalttransistor BC107B oder ähnliche
1	Diode 1N4148
1	Leuchtdiode, Farbe nach Wahl
1	Widerstand 10 Kiloohm
1	Widerstand 330 Ohm

Tabelle 2.16: Die Einzelteile des Teleschalters

2.3.4.4 Die Teleschalter-Software

Das eigentliche Steuerprogramm liegt als Maschinenprogramm TELESCHALTER. OBJ vor. Es belegt den Bereich \$8000 – \$8176 und besitzt einen Autostart, so daß es mit der Reset-Taste gestartet werden kann. Ebenso geht das auch mit SYS 32806 (= \$8026).

In der Originalversion des Maschinenprogramms sind folgende Funktionen festgelegt:

Um eine Aktion auslösen zu können, müssen zuerst vier Impulse ankommen, es muß also beispielsweise viermal klingeln. Wenn es danach innerhalb von etwa zehn Minuten noch einmal klingelt, schaltet sich das rechte Relais Nummer 1 ein. Klingelt es zuerst viermal und dann zweimal, schaltet sich Kanal 2 ein, bei dreimal Kanal 3 und bei viermal Kanal 4.

Klingelt es zuerst viermal und danach fünfmal, dann wird das Relais 1 ausgeschaltet, analog bei sechsmal Kanal 2, bei siebenmal Kanal 3 und bei acht Impulsen Kanal 4. Ein besonderer Leckerbissen bietet sich, wenn es beim zweiten Anruf neunmal klingelt. Danach nämlich wird ein Maschinenunterprogramm bei \$C000 (= 49152 dezimal) aufgerufen, das vom Anwender selbst geschrieben werden kann. Damit sind der Phantasie keine Grenzen mehr gesetzt. Wenn man mit dem Autoreset arbeitet, darf nicht vergessen werden, im eigenen Programmteil genügend oft die Antireset-Impulse an $\overline{PC2}$ auszulösen. Denkbar wäre zwar die Erledigung dieser Aufgabe in einer Interrupt-Routine, es ist jedoch möglich, daß eine solche Routine auch weiterläuft, wenn der Rechner bereits unplanmäßig das Hauptprogramm verlassen hat. In diesem Fall wäre die Autoresetschaltung sinnlos...

Die Forderung, daß eine Reaktion erst beim zweiten Anruf erfolgt, wurde ebenfalls aus Gründen der Störsicherheit ein-

geführt. Man muß damit rechnen, daß ahnungslose Fremde anrufen und genau dann frustriert den Hörer auflegen, wenn der Zähler ausgerechnet den richtigen Stand hat. Ein Zufall – sicher, aber daß innerhalb der eingestellten Zeit zwei Anrufe kommen, die jeweils genau die richtigen Klingelzahlen haben, das läßt sich schon mit an Sicherheit grenzender Wahrscheinlichkeit ausschließen. Es genügt nicht, daß mindestens die eingestellten Klingelimpulse ertönen, sondern ihre Zahl muß haargenau stimmen. Das Steuerprogramm wartet die vorbestimmte »Maximalzeit zwischen zwei Impulsen« ab, und erst wenn danach Ruhe eingekehrt ist, springt es aus dem Unterprogramm ZAEHLEN zurück und löst eventuell vorgesehene Aktionen aus.

Um das Steuerprogramm TELESCHALTER.OBJ nach eigenen Ansprüchen modifizieren zu können, wird das BASIC-Programm TELEPATCH verwendet. Hier kann man nach Herzenslust einstellen, wieviele Klingelimpulse die Aktionen einleiten sollen, wieweit die Impulse auseinander liegen dürfen, innerhalb welcher Zeit der zweite Anruf erfolgen muß, wieviele Klingelimpulse welche Aktion auslösen und was bei welcher Aktion ein- oder ausgeschaltet werden soll, ob und wo ein eigenes Programm aufgerufen werden soll, mit welcher Versionsnummer sich das Steuerprogramm melden soll und schließlich, wo es im Computerspeicher liegen soll. (Autostart ist nur bei $8000 möglich!)

Man kann die Parameter sogar so einstellen, daß bereits nach dem zweiten Impuls das eigene Programm startet. In diesem Fall wird gleichzeitig Kanal 1 eingeschaltet. Diese Betriebsart ist praktisch für die mechanische Steuerung einer eigenen Mailbox. Dazu wird der Telefonhörer abgenommen und fest in die Muscheln des Akustikkopplers (zum Beispiel aus Kapitel 2.6.3) gelegt. Normalerweise drückt ein Gewicht die Gabel des Telefons herunter und simuliert somit einen aufgelegten Handapparat. Klingelt es nun, wird über das Relais 1 ein Motor eingeschaltet, der das Gewicht hochhebt. Gleichzeitig wird das Mailboxprogramm aufgerufen. Zur Realisierung der Mechanik eignen sich Fischertechnik-Baukästen sehr gut. Daß zwei Klingelimpulse zum Ansprechen nötig sind, wurde wiederum aus Gründen der Störsicherheit gewählt. Es kann vorkommen, daß die Schaltung schon auf lautes Türzuschlagen anspricht. In unserem Fall reicht das noch nicht aus für eine Aktion. Andererseits soll nicht zuviel Zeit bis zum Zustandekommen der Verbindung vergehen.

Durch ein und dieselbe Klingelfolge können auch mehrere unterschiedliche Aktionen ausgelöst werden.

Das BASIC-Programm lädt den Original-Maschinencode TELESCHALTER.OBJ nach und speichert zum Schluß die gepatchte Version auf Diskette ab.

2.3.5 Der C64 lernt sprechen

2.3.5.1 Ein Sprachsynthesizer

Nichts Besonderes, werden Sie denken. Wohl jeder C64-Anwender hat schon einmal den Software-Sprachsynthesizer SAM plaudern hören. Das ist für sich betrachtet schon ganz lustig, doch der Spaß hört auf, sobald man SAM auch in eigenen Programmen einsetzen will. Spätestens dann fällt einem nämlich auf, daß der Rechner

während der Sprachausgabe vollauf beschäftigt ist und gar nichts anderes mehr kann, ja er schaltet sogar das Bild ab!
Das ist bei der hier vorgestellten Hardwarelösung vollkommen anders. Das Herz dieser Sprachausgabekarte bildet der Baustein SC-01 vom amerikanischen Hersteller Votrax. Es handelt sich um einen CMOS-Chip, der beliebige Worte aus einzelnen Lauten (Phonemen) zusammensetzen kann. Mit den sechs Eingangsleitungen P0 bis P5 kann der Anwender jeweils eins aus 64 Phonemen auswählen, das dann in einer von drei möglichen Betonungsstufen ausgegeben wird. Die erzielbaren Laute zeigt Tabelle 2.17 zusammen mit ihrer spezifischen Dauer. Da es sich um einen amerikanischen Baustein handelt, sind englische Klangbeispiele angegeben.

Phonem Hex	Code Dez	Dauer in ms	Beispiel
$00	0	59	jacke!t
$01	1	71	e!nlist
$02	2	121	hea!vy
$03	3	47	(kein Laut)
$04	4	47	butt!er
$05	5	71	ma!de
$06	6	103	ma!de
$07	7	90	az!ure
$08	8	71	ho!nest
$09	9	55	inhibi!t
$0A	10	80	i!nhibit
$0B	11	121	inhi!bit
$0C	12	103	m!at
$0D	13	80	sun!
$0E	14	71	b!ag
$0F	15	71	v!an
$10	16	71	ch!ip
$11	17	121	sh!op
$12	18	71	z!oo
$13	19	146	law!ful
$14	20	121	thing!
$15	21	146	fa!ther
$16	22	103	loo!king
$17	23	185	boo!k
$18	24	103	l!and
$19	25	80	trick!
$1A	26	47	j!udg!e

Phonem Hex	Code Dez	Dauer in ms	Beispiel
$1B	27	71	h!ello
$1C	28	71	g!et
$1D	29	103	f!ast
$1E	30	55	paid!
$1F	31	90	pass!
$20	32	185	day!
$21	33	65	day!
$22	34	80	y!ard
$23	35	47	missio!n
$24	36	250	mo!p
$25	37	103	pa!st
$26	38	185	co!ld
$27	39	185	pi!n
$28	40	185	mo!ve
$29	41	103	any!
$2A	42	71	t!ap
$2B	43	90	r!ed
$2C	44	185	mee!t
$2D	45	80	w!in
$2E	46	185	da!d
$2F	47	103	a!fter
$30	48	90	sa!lty
$31	49	71	a!bout
$32	50	103	u!ncle
$33	51	185	cu!p
$34	52	80	fo!r
$35	53	121	aboa!rd
$36	54	59	you!
$37	55	90	you!
$38	56	80	th!e
$39	57	71	th!in
$3A	58	146	bi!rd
$3B	59	185	ge!t
$3C	60	121	be!
$3D	61	250	ca!ll
$3E	62	185	(kein Laut)
$3F	63	47	(kein Laut)

Tabelle 2.17: Phoneme des SC-01

Die Phoneme des SC-01 sind hauptsächlich für die englisch-amerikanische Sprache eingestellt. Es läßt sich trotzdem eine gute deutsche Aussprache erzielen, wenn die folgenden Hinweise beachtet werden: Nicht alle Phoneme sind für die deutsche Sprache uneingeschränkt brauchbar. Einzelne in der deutschen Sprache vorkommende Laute muß man umgehen oder überspielen. Die Hauptschwierigkeit liegt bei den Vokalen. Für möglichst natürlichen Klang sollten die in Tabelle 2.18 folgenden Phoneme benutzt werden.
Wenn Konsonanten schärfer klingen sollen, empfiehlt es sich, eine kurze Pause (Phonem 3) voranzustellen. Durch Anfügen können auf diese Weise auch Endkonsonanten verstärkt werden (zum Beispiel Weg). Ein End-D kann als DT verstärkt werden. Wenn innerhalb eines Wortes

A	(lahm)	21, 8 (unter Umständen beide hintereinander)
A	(ab)	21
Ä		48, 21, 1
E	(Lehm)	33, 6
E	(Lebe!n)	6 (0 auch möglich)
I	(lieb)	41 – 41 (zweimal)
I	(Licht)	41, 39
O	(Lob)	38 (unter Umständen zweimal), 22
O	(ob)	53
Ö	(zwölf)	58 (nicht immer darzustellen)
U	(Ruf)	55 (unter Umständen zweimal)
U	(und)	55, 22, 45
Ü	(für, hübsch)	10 (schlecht darstellbar)
EU	(äu)	53 – 00 – 41
EI		21 – 41
AU		8 – 35 oder 21 – 35 -45
CH	(ich, ach)	27 – 27, unter Umständen 57, 16
R		43 (klingt amerikanisch)
ER	(Herr, lieber)	8 (verschluckt gesprochen)
C,Z	(Cäsar, Zeit)	42 – 31
J		41
S	(sanft)	18
ß		31
V	(viel)	29
W		15
SCH		17
NG	(eng)	20
G	(Genie)	26

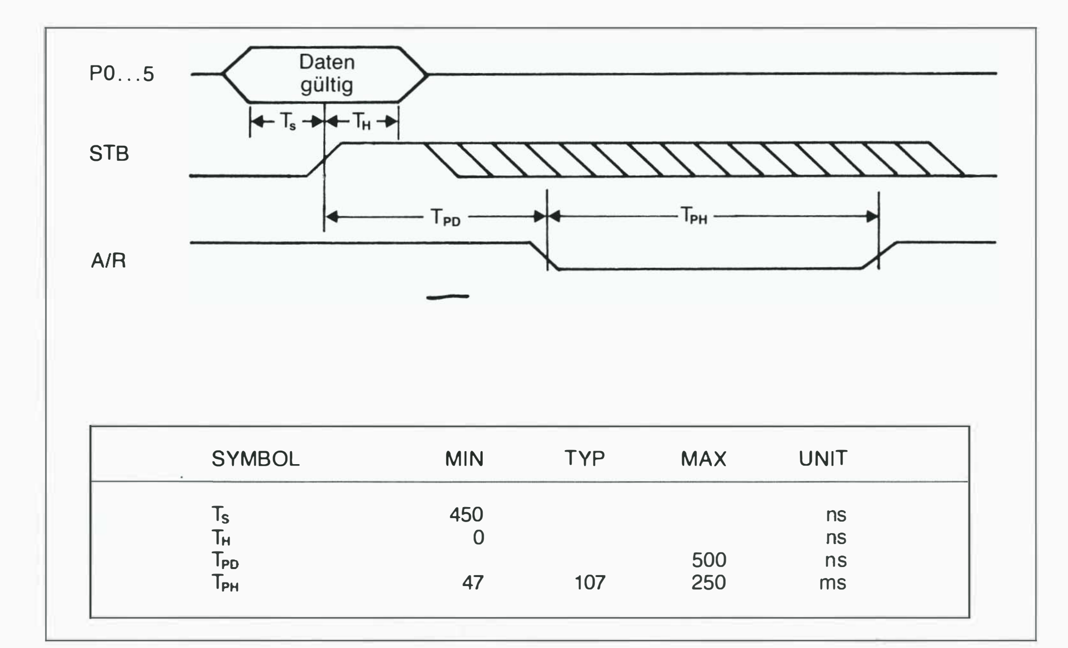
Tabelle 2.18: Phoneme speziell für deutsche Sprache (Angaben dezimal)

mehrere Konsonanten nacheinander stehen, ist es ratsam, sie durch eine Pause zu trennen, damit sie klarer klingen. Auch sollten lange Worte sinngemäß und nach Silben durch eine kurze Pause getrennt werden.

Zur Betonung: Es sind meist nur ein oder zwei Stufen benutzbar. Betonungsänderungen von 0 auf 1 oder 2 sollten – wenn möglich – nicht direkt vor Vokalen vorgenommen werden, sondern schon vor stimmhaften, stimmlosen oder mit Pausen versehenen Konsonanten. Wenn Betonungsänderungen direkt vor dem Vokal oder zwischen zwei Vokalen vorgenommen werden, tönt es wie Stimmbruch oder Jodeln.

2.3.5.2 Die Ansteuerung des Sprachsynthesizers

Bild 2.49 zeigt das Zeitdiagramm zur Ausgabe eines Lautes beim SC-01. Das ausgewählte Phonem wird an die Leitungen P0 bis P5 angelegt und dann mit STB (Strobe) der Befehl zum Aussprechen gegeben. Der Sprachbaustein enthält einen Zwischenspeicher, der intern aus sechs parallel geschalteten D-Flipflops besteht – für jede Leitung eins. Bei dieser Gelegenheit soll einmal die Funktionsweise von D-Flipflops genau erläutert werden. Bereits in Kapitel 1.6 wurde die Schaltung eines solchen Speicherelements gezeigt und die Funktionsweise kurz angesprochen. Der Takteingang C ist hier mit STB verbunden. Die Daten werden genau zu dem Augenblick übernommen, in dem STB von LOW nach HIGH geht, jedoch müs-



SYMBOL	MIN	TYP	MAX	UNIT
T_S	450			ns
T_H	0			ns
T_{PD}			500	ns
T_{PH}	47	107	250	ms

Bild 2.49: Zeitdiagramm zum Datenaustausch mit dem SC-01 und die zugehörigen Zeiten

sen sie während eines bestimmten Zeitintervalls unverändert anliegen. Vor der Datenübernahme darf sich während der sogenannten Setup-Zeit (T_S) nichts mehr ändern und auch nach der positiven Strobe-Flanke müssen die Daten noch für die Dauer der Hold-Zeit (T_H) stabil bleiben. Danach haben Änderungen der Datenleitungen keinerlei Einfluß mehr auf den Speicherzustand, bis wieder eine positive STB-Flanke auftritt.
Die Setup- und Hold-Zeiten sind bei allen Speicherelementen zu beachten und in den einschlägigen Datenblättern angegeben. Natürlich kommt es immer auf das jeweilige Speicherelement an, ob die Daten während der positiven oder negativen Taktflanke übernommen werden.

2.3.5.3 Zeit im Diagramm

Im Zeitdiagramm (Bild 2.49) wird der beschriebene Vorgang symbolisch deutlich gemacht. Alle sechs Datenleitungen, P0 bis P5, wurden in eine Zeile eingetragen. Der Strich in der Mitte dieser Zeile, also zwischen HIGH und LOW, sagt aus, daß momentan der Zustand dieser Leitungen für die Funktion unerheblich ist. Nur während des Abfrageintervalls $T_S + T_H$ müssen feste Zustände eingehalten werden. Jede Leitung muß während dieser Zeit durchgehend entweder HIGH oder LOW sein. Je nachdem, welche Datenkombination angelegt wird, können einige Leitungen HIGH, andere aber LOW sein. Während der Abschnitte mit schrägen Begrenzungslinien stabilisieren sich die Spannungen noch, doch dann entsteht im Zeitdiagramm ein Fenster, in dem keine Übergänge mehr auftreten.

In der zweiten Zeile des Diagramms ist das Strobe-Signal (STB) eingezeichnet. Der Zeitpunkt des LOW-HIGH-Wechsels ist fest, doch ist es egal, wann die Rückkehr nach LOW erfolgt. Daher wird ein Übergangsraum angegeben, der durch das schraffierte Feld symbolisiert wird.
Die dritte Zeile gibt schließlich ein Ausgangssignal des Sprachbausteins an. Es reagiert auf die Datenübernahme nach einer gewissen Verzögerungszeit T_{PD} (Propagation Delay) mit einer negativen Flanke und kehrt erst wieder nach HIGH zurück, wenn der Laut vollständig ausgesprochen ist und der nächste übernommen werden kann. Die LOW-Zeit entspricht also der Dauer des ausgesprochenen Phonems T_{PH}.
Sieht man sich die zugehörigen Zeiten an, dann fällt auf, daß dieses Impulsdiagramm ganz und gar nicht maßstäblich ist. Es dient nur zur Verdeutlichung des Ablaufs und gibt an, zwischen welchen Punkten die Zeiten gemessen wurden.

2.3.5.4 Schon wieder Handshake

Das Zeitdiagramm macht bereits deutlich, wie der Baustein am sinnvollsten anzuschließen ist. Es handelt sich um einen klassischen Handshake-Fall. STB könnte direkt von $\overline{PC2}$ bedient werden und A/R das $\overline{FLAG}$-Bit setzen, wenn nicht... ja, wenn nicht die Signale genau invertiert vorlägen! Dazu kommt noch, daß der CMOS-Chip SC-01 mit 9 Volt betrieben wird. Intern ist zwar dafür gesorgt, daß die Eingänge TTL-kompatibel sind, doch die Spannung des Ausgangs A/R muß begrenzt werden. Der Schaltplan (Bild 2.50) zeigt die Lösung des Problems. Wir setzen

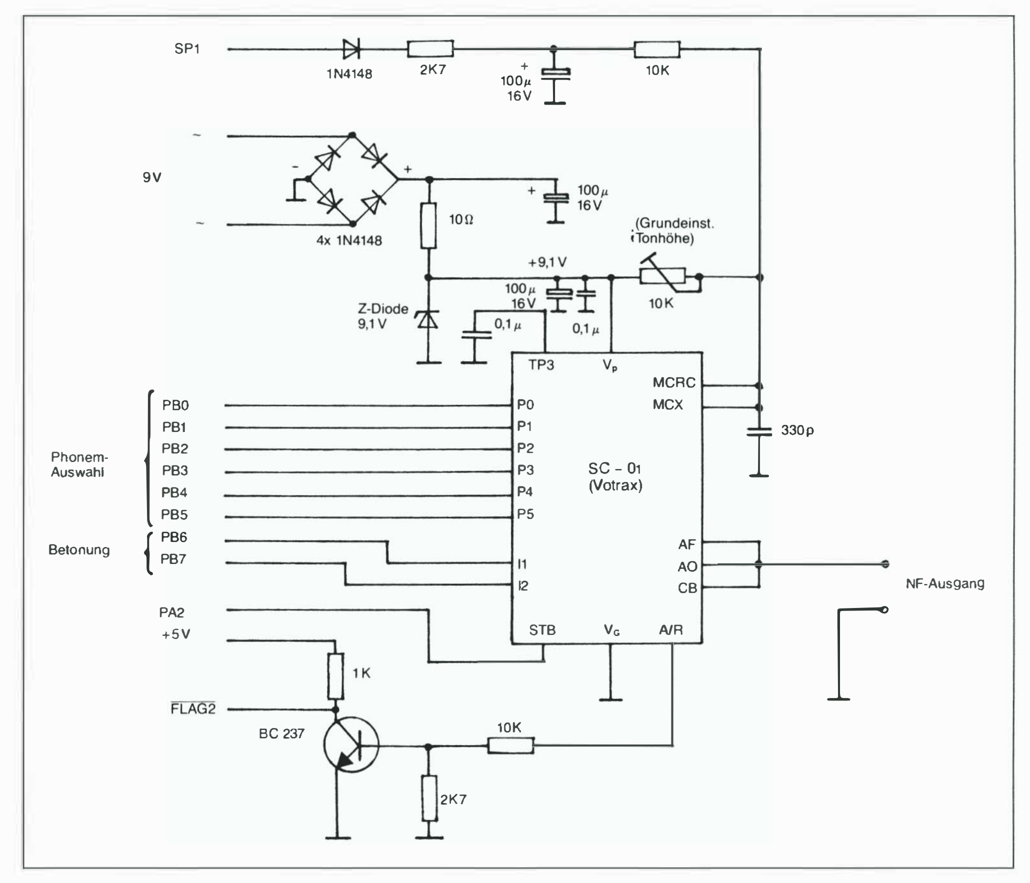


Bild 2.50: Schaltung der Sprachausgabekarte (ohne Verstärker)

zwischen den A/R-Ausgang und den Anschluß $\overline{\text{FLAG2}}$ eine Transistorstufe, die einerseits die Spannungsamplitude an die Erfordernisse des C64 angleicht und andererseits den logischen Pegel invertiert. Das Strobe-Signal erzeugen wir der Einfachheit halber mit der freien Portleitung PA2.

Bleibt noch die Spannungsversorgung des SC-01, doch auch für solche Probleme ist der C64 gerüstet. Am User-Port liegt ja nicht umsonst auch eine Wechselspannung von 9 Volt. Sie wird von vier Universaldioden in Brückenschaltung gleichgerichtet, über einen Elko geglättet und in einem Spannungsteiler mit einer Zenerdiode auf 9,1 Volt stabilisiert. Zwei Kondensatoren verhindern nochmals Störeinwirkungen durch Stromschwankungen beim Sprechen von scharfen Konsonanten. Über den Trimmer kann die Grundtonhöhe des Bausteins reguliert werden.

Mit einer additiven Spannung an den Anschlüssen MCRC und MCX des SC-01 kann die Tonhöhe der Sprachausgabe in gewissen Grenzen verändert werden. Er-

zeugt man diese Spannung vom Computer, dann ist es möglich, die Tonhöhe per Programm zu variieren. Diese Möglichkeit sollte man doch ausnutzen, aber wie läßt sich allein mit den vorhandenen User-Portanschlüssen eine Gleichspannung erzeugen? Nun – in der Schaltung wird bereits der serielle Daten-Port benutzt, der erst später in diesem Buch an der Reihe ist. In Kapitel 2.5.2 wird auf dieses Problem nochmals eingegangen.

2.3.5.5 Aufbau der Sprachausgabekarte

Bild 2.51 zeigt das Layout der Sprachausgabeplatine. Wie Sie sehen, besteht sie aus zwei Teilen. Auf der Seite für den noch näher besprochen. Falls Sie den Sprachsynthesizer lieber an einen anderen Verstärker anschließen wollen, beispielsweise an Ihre Stereoanlage oder an einen NF-Eingang Ihres Monitors, dann können Sie die Platine an der angedeuteten mittleren Begrenzungslinie durchtrennen.

Die Bestückung der beiden Platinen nehmen Sie bitte nach Bild 2.52, Seite 112, vor. Beachten Sie dabei die allgemeinen Hinweise in Anhang A. Tabelle 2.19 führt alle benötigten Teile auf.

Der User-Port-Stecker wird mit seiner unteren Kontaktleiste auf der Platinenunterseite verlötet. Drei Pins der oberen Leiste werden mit den gekennzeichneten Punkten direkt verbunden, ein vierter über die

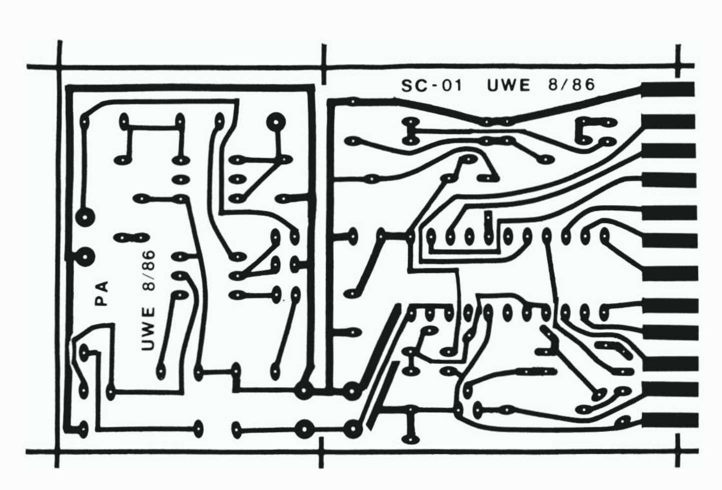


Bild 2.51: Die Platinenvorlage für Sprachausgabe und Verstärker

User-Port-Stecker befindet sich die gerade besprochene Schaltung mit dem SC-01. Der hintere Teil dagegen enthält einen Leistungsverstärker, an den direkt ein Lautsprecher angeschlossen werden kann. Die Verstärkerschaltung wird im Kapitel 5.3 Diode. Falls der TBA 800 versetzte Anschlußbeinchen hat, müssen diese vorsichtig mit einer Flachzange gerade gebogen werden, um in den IC-Sockel zu passen.

Foto 2.7 zeigt die fertig bestückte Platine mit Sprachausgabe- und Verstärkerteil.

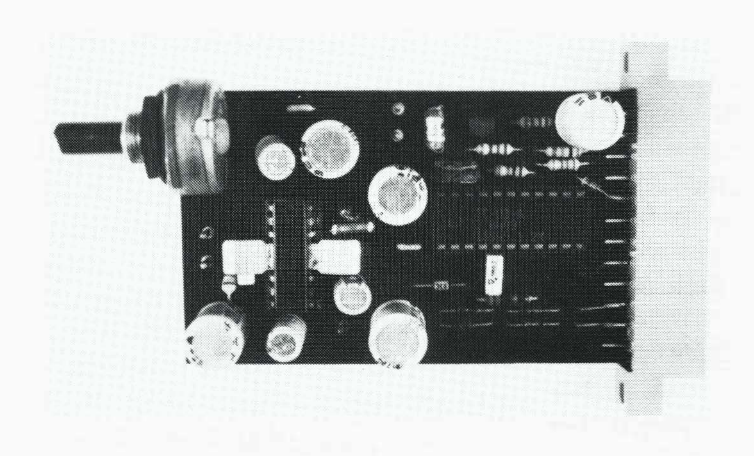


Foto 2.7: Sprachausgabe und NF-Verstärker betriebsfertig

Sprachausgabe:	Verstärkerteil:
1 Sprachsynthesizer SC-01 (z.B. Digital Service, Landwehrstraße 37, 8000 München 2)	1 NF-Verstärker-IC TBA 800
1 Sockel 22pol	1 Sockel 16-pol
1 Transistor BC237 oder ähnliche	1 Widerstand 51 Ohm
1 Zenerdiode 9,1 Volt (ZPD 9V1)	1 Widerstand 1,2 Ohm
5 Dioden 1N4148	1 Poti 100 Kiloohm, logarithmisch
2 Widerstände 10 Kiloohm	1 Kondensator 220 Mikrofarad / 16 Volt, radial
2 Widerstände 2,7 Kiloohm	4 Kondensatoren 100 Mikrofarad / 16 Volt, radial
1 Widerstand 1 Kiloohm	2 Kondensatoren 0,1 Mikrofarad
1 Widerstand 10 Ohm	1 Kondensator 3 Nanofarad
1 Trimmpotentiometer 10 Kiloohm	1 Kondensator 70 Pikofarad
3 Elektrolytkondensatoren 100 Mikrofarad / 16 Volt	1 Lautsprecher
2 Kondensatoren 0,1 Mikrofarad, Keramik	bei Einzelaufbau:
1 User-Port-Stecker	1 einseitige Platine nach Bild 2.51 (Verstärkerteil)
1 einseitige Platine nach Bild 2.51	

Tabelle 2.19: Einzelteile der Sprach- und Verstärkerplatine

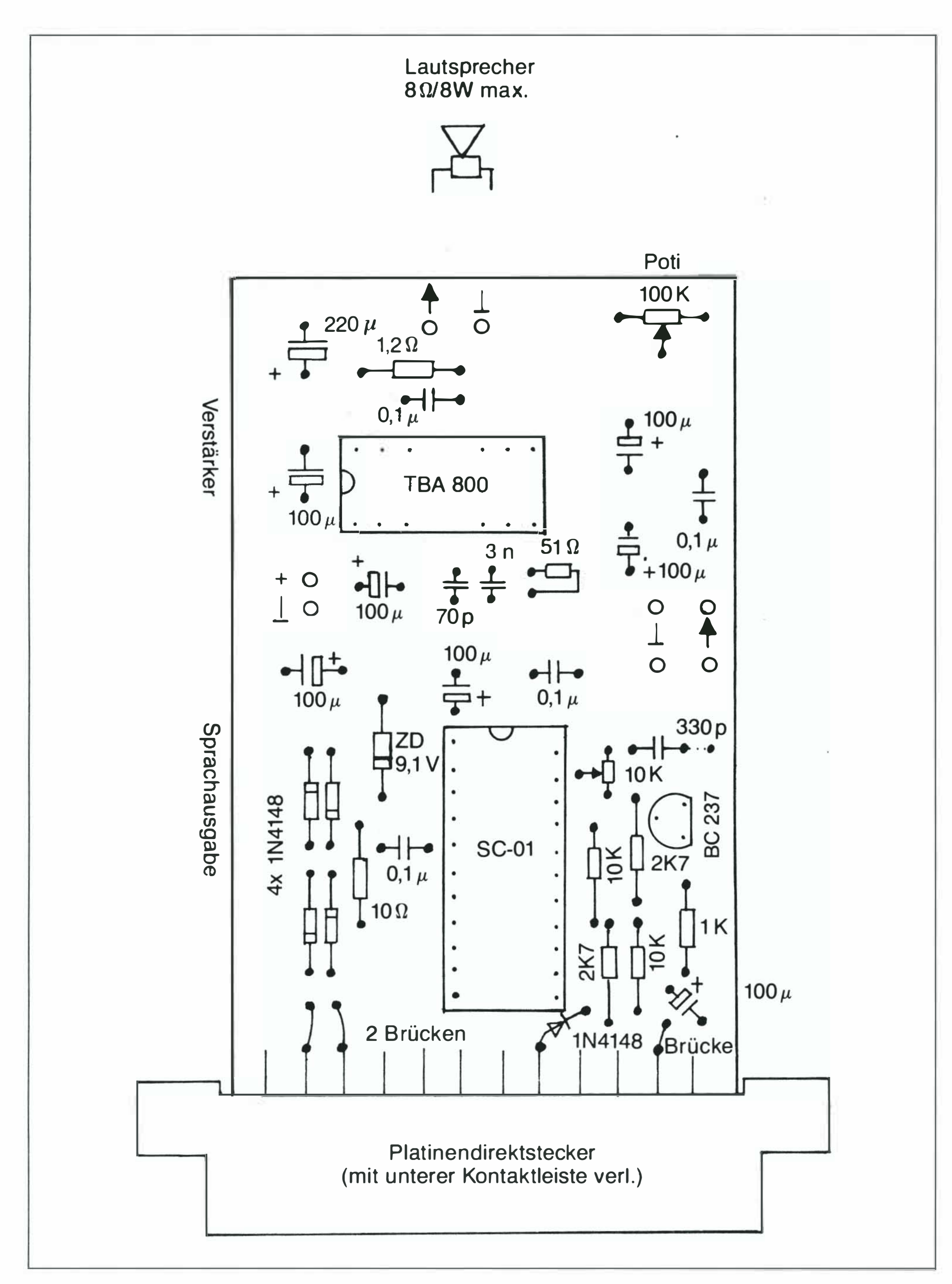


Bild 2.52: Bestückungsplan der Sprachausgabekarte und des Verstärkers

2.3.5.6 Probesprechen

Das kurze Programm SC-01 DANK zeigt, wie der Sprachbaustein von BASIC aus zu bedienen ist. In den DATAs ist ein kurzes Textbeispiel abgelegt. Es wird Zeichen für Zeichen gelesen und auf den Port ausgegeben. Danach wird jeweils PA2 auf HIGH und gleich anschließend wieder auf LOW gesetzt (STB-Impuls). Nun wartet der Computer in einer Schleife das Setzen des FLAG-Bit in CIA # 2 ab, um den nächsten Laut auszugeben.

Natürlich ist es sehr umständlich und zeitaufwendig, jeweils die passenden Phoneme aus der Liste herauszusuchen und als DATA-Zeile hintereinanderzusetzen. Daher wurde ein kleiner »Sprachcompiler« entwickelt, der schriftlich eingegebenen deutschen Text in eine Bytekombination umwandeln und auch aussprechen kann. Er befindet sich als SC-01 COMPILER auf der Diskette.

Ins Auge fällt sofort die Phonem-Liste, in der alle Laute mit ihrer im deutschen Sprachraum üblichen Schreibweise in das Feld T$ und alle zugehörigen Phoneme oder Phonemkombinationen in M$ eingeschrieben werden. Für viele Schreibweisen gibt es mehrere Klangalternativen. Der Sprachanalytiker nennt sie Mehrdeutigkeiten. M$ ist daher ein zweidimensionales Feld, wobei die Alternativen über die zweite Feldvariable (im Programm M) ausgewählt werden.

Bei der Texteingabe entsteht der Eingabestring E$. Zunächst werden drei Zeichen als Suchkette S$ daraus isoliert und mit den gespeicherten Schreibweisen verglichen. Dies geschieht, um Laute wie SCH berücksichtigen zu können, die trotz ihrer drei Buchstaben nur als ein einziger Laut ausgesprochen werden. Findet sich die Kombination nicht, reduziert das Programm die Kette auf zwei und bei einem weiteren Durchgang gegebenenfalls nochmals auf ein Zeichen. Existiert jedoch die Suchkette auch in der Phonemliste, dann wird der dazu gespeicherte Code in den Ablagespeicher übernommen.

Normalerweise arbeitet das Programm mit der Betonungsstufe 0 (B=0) und der Grundaussprache (Mehrdeutigkeit M=1). Findet es jedoch eine andere Betonungsziffer, dann werden alle folgenden Phoneme mit der zu ihr gehörenden Betonungsbit-Kombination abgelegt. Bei den Mehrdeutigkeitskennzeichen ».« und »–« erfolgt nur die Auswahl des nächsten Lautes mit der angegebenen Aussprache. Wählt man eine Alternative, zu der keine Daten eingegeben sind, wird die nächstkleinere Aussprachemöglichkeit übernommen. Statt -EI (M=3: nicht vorhanden) wird also .EI (M=2: 21 41) abgelegt.

Um bei der Umsetzung nicht zu lange Wartezeiten in Kauf nehmen zu müssen und um die Sprachausgabe flott ablaufen zu lassen, empfiehlt sich die Bearbeitung des Programms mit einem BASIC-Compiler. Auf der beiliegenden Diskette ist neben dem BASIC-Quelltext eine solche kompilierte Version unter dem Namen SC-01 COMPILER/C vorhanden.

Falls Sie mit einigen Feinheiten der Software nicht zufrieden sind, können (und sollten) Sie auch Änderungen am BASIC-Quelltext vornehmen.

Eine Möglichkeit besteht in der Verlängerung der Phonemliste. Dabei ist zu beachten, daß längere Zeichenketten immer vorne in der Liste auftauchen sollten. Die

Länge der Liste wird gleich zu Anfang in L angegeben (Zeile 504).
Eine weitere Änderungsmöglichkeit wäre das Einfügen von speziellen Sprachregeln nach Zeile 4000. Dort wurde bereits viel Platz gelassen.
Damit Sie das Programm leichter durchschauen können, hier eine Liste der verwendeten Variablen und ihre Bedeutung:

B	Betonung
G	Gerätenummer (Diskette 8, 9 / Kassette 1)
I, J, K	Laufvariablen
L	Länge der Phonemliste
LS	Länge der Suchzeichenkette
M	Mehrdeutigkeitszeiger
R	Speicherzeiger zum Ablegen des Phonemwertes
RA	Phonemspeicher-Anfang
RE	Vorläufiges Ende der Phonemkette
A$	Tastatureingabe
E$	Eingabezeichenkette
N$	Filename
P$	Phonemkette des einzelnen Lautes
S$	Suchzeichenkette
M$(...,1)	1. Aussprachemöglichkeit
M$(...,2)	2. Aussprachemöglichkeit
M$(...,3)	3. Aussprachemöglichkeit
T$( )	Teilzeichenkette (1, 2 oder 3 Buchstaben)

Tabelle 2.20: Variablenliste vom SC-01-Compiler

2.3.5.7 Bedienung des Sprachcompiler

Beim Starten des Programms SC-01 COMPILER bzw. SC-01 COMPILER/C werden Sie zunächst gefragt, wo die erzeugte Phonemkette im Speicher abgelegt werden soll. Drücken Sie nur RETURN, beginnt der Computer bei $C000, also bei dezimal 49152. Dort sind bekanntlich 4 Kilobyte RAM-Speicher für eigene Anwendungen freigehalten, was auch für überlange Texte in jedem Fall ausreichen dürfte. Es folgen einige Erläuterungen und die Aufforderung, Text einzugeben. Schreiben Sie etwas hinter das Fragezeichen, zum Beispiel: GUTEN TAG. Drücken Sie RETURN und warten Sie einen Moment. Der Computer wird nach der Umsetzung sofort den Text vorsprechen. Danach erscheint ein Menü auf dem Bildschirm, in dem zwischen etlichen Funktionen ausgewählt werden kann.

Drücken Sie einmal A, um den gerade eingegebenen Text zu ändern. Die Buchstaben erscheinen wieder, und Sie können zum Beispiel Betonungsstufen hinzufügen. Einfügen ist genau wie beim normalen BASIC-Editor möglich. Wie wäre es

mit 2GUTEN 0TAG ? Alles hinter einer der Ziffern 0, 1 oder 2 wird mit der jeweiligen Betonungsstufe ausgesprochen, bis die nächste Ziffer folgt.

Viele Buchstaben klingen in verschiedenen Worten ganz anders. Daher wurden im Programm oft Alternativen festgelegt, die durch Voranstellen von ».« oder »–« ausgewählt werden können. Ein häufig vorkommendes Beispiel dafür ist der Buchstabe E. Nehmen Sie das Wort EBEN. Das erste E wird deutlich und hoch ausgesprochen, das zweite dagegen vielleicht mehr als kurzes Ö. Hört man genau hin, kommt das zweite E aber öfter vor. Es wurde daher als Normalphonem festgelegt. Wollen Sie das erste E erreichen, so müssen Sie .E eingeben. Die richtige Ausgabe erhält man also bei .EBEN.

Ein weiteres Beispiel ist das ST, das ja als ein Laut behandelt wird. Es klingt normalerweise wie in STEIN. Soll ein spitzes ST wie in KISTE ertönen, so müssen Sie auch hier einen Punkt voranstellen, also KI.STE schreiben.

Im allgemeinen wird ohne Zusätze der kürzeste Laut ausgegeben, mit vorangestelltem Punkt eine etwas längere Alternative und bei vorangestelltem »–« die längste Möglichkeit. Vokale lassen sich auch durch Verdoppelung verlängern.

Wenn Ihnen der so zusammengesetzte Textabschnitt gefällt, dann drücken Sie W, um weiteren Text anzuhängen. Dies geschieht nach denselben Richtlinien. Mit Z wird jeweils die zuletzt eingegebene Textzeile noch einmal vorgesprochen, mit G der gesamte Text. Oft soll zwischen zwei Eingaben eine Pause entstehen. Dazu geben Sie am Anfang der Zeile eine Betonungsziffer ein und lassen einige Leerzeichen folgen. Die Betonungsziffer ist wichtig, denn andernfalls werden führende Leerzeichen ignoriert. Ebenso verhält es sich sinngemäß beim Abschließen nachgestellter Leerzeichen mit einer Betonungsziffer am Ende der Eingabezeile.

Gute Texte sollten Sie speichern. Drücken Sie dazu S und geben Sie nach der Aufforderung einen Filenamen an. Danach können Sie die Gerätenummer wählen. Bei Kassettenbetrieb tippen Sie 1 und RETURN, nur RETURN aktiviert das Diskettenlaufwerk mit der Gerätenummer 8.

Die Phonemkette wird wie ein Maschinenprogramm gespeichert und kann nach Ausstieg aus dem Programm wieder mit ,8,1 absolut geladen werden.

Auf diese Weise kann eine Bibliothek vorgefertigter Textelemente angelegt werden. Mit der Funktion L lassen sich diese Teile beliebig zusammenfügen, indem der Text mit dem angegebenen Filenamen an den aktuellen Inhalt des Speichers angehängt wird. Sollten Sie aus Versehen einen falschen Text geladen haben, drücken Sie A für Ändern und geben anstelle des Textes nur RETURN ein. Der zuletzt angehängte Text ist damit gelöscht, und Sie können einen neuen Versuch starten. Auch hier spricht Ihnen das Programm natürlich mit Z den zuletzt angehängten und mit G den gesamten Text vor.

Mit P können Sie sich informieren, wieviel Platz schon von der Phonemkette belegt wurde. Sie werden sich wundern, wie gering der Speicherbedarf ist.

2.3.5.8 Sprache aus dem Hintergrund

Wie gesagt, spricht der SC-01 zum größten Teil selbständig. Die Zeit, um ihm

das zu sprechende Phonem mitzuteilen, ist sehr sehr kurz im Vergleich zur Sprechdauer. Außerdem meldet er über das Flag-Interrupt-Bit selbständig, wenn er ein Phonem fertig ausgesprochen hat. Damit schreit die Schaltung geradezu nach Interrupt-Bedienung im Hintergrund. Natürlich kann eine effiziente Interrupt-Routine nur auf der Maschinenebene ablaufen. Das kurze Assembler-Programm ist aber auch für Maschinensprache-Anfänger leicht verständlich.

Machen wir uns einmal klar, was passiert, wenn der SC-01 einen Laut fertig ausgesprochen hat:

Der Anschluß A/R ist ja während der Aussprache eines Lautes LOW. Es tritt also eine steigende Flanke auf, die durch den Transistor invertiert wird, und das $\overline{\text{Flag}}$-Bit im Interrupt-Kontrollregister von CIA # 2 setzt. Bei geeigneter Programmierung des CIA-Bausteins geht sein Anschluß $\overline{\text{IRQ}}$ infolgedessen auf LOW. Diese Leitung wirkt auf den $\overline{\text{NMI}}$-Anschluß des Prozessors und erzeugt damit einen nicht maskierbaren – das heißt softwaremäßig nicht abschaltbaren – Interrupt.

Ein Interrupt ist eine Unterbrechung des Programmablaufs. Beim Eintreffen einer negativen Flanke am Anschluß $\overline{\text{NMI}}$ bearbeitet der Prozessor zunächst den laufenden Befehl zu Ende, legt dann – wie bei einem Unterprogrammaufruf – die augenblickliche Arbeitsposition auf den Stack und holt sich aus den Adressen $FFFA und $FFFB die Startadresse der Interrupt-Routine ab, mit der er fortfährt. Anders als bei einem Unterprogramm muß am Ende einer Interrupt-Routine der Befehl RTI (Return From Interrupt = Rückkehr von der Unterbrechung) stehen.

Der C64 bietet glücklicherweise die Möglichkeit, von der eigentlich fest im ROM abgelegten Interrupt-Routine aus eigene Programmteile anzuspringen. Beim Original-ROM ist die Startadresse der NMI-Routine nämlich $FE43. Dort werden zunächst die maskierbaren Interrupts (IRQ) gesperrt (mit SEI), und dann folgt ein Sprung über den NMI-Vektor im RAM ($0318/$0319 = dezimal 792/793), der normalerweise wieder zurück auf $FE47 zeigt, also eigentlich wirkungslos ist. Wird nun dieser Vektor verbogen, das heißt, ändert man die Sprungadresse im RAM – was ja im Gegensatz zum ROM möglich ist – kann man zu eigenen Routinen verzweigen. Genau hier setzt unsere Routine ein.

Sehen Sie sich das Programm einmal an. Zunächst werden alle Register auf den Stack gelegt, damit ihre Werte bei der Rückkehr zum unterbrochenen Hauptprogramm nicht verändert sind und alles durcheinander bringen. Dann wird das Interrupt-Kontrollregister abgefragt und getestet, ob der Interrupt tatsächlich vom Flag-Bit erzeugt wurde. Falls das nicht der Fall gewesen sein sollte, muß ein ganz normaler NMI-Impuls aufgetreten sein, zum Beispiel durch Drücken der RESTORE-Taste, die ja ebenfalls auf diesen Prozessoreingang wirkt, und es erfolgt ein Rücksprung zur alten NMI-Routine im ROM, natürlich nach der Rettung der Register auf den Stack.

Bei gesetztem Flag-Bit jedoch wird über einen Zeiger das aktuelle Phonembyte geholt. Falls es den Wert $FF (255) hat, erkennt die Routine das Ende der Phonemkette und beendet die Ausgabe. Ansonsten wird das Byte auf den User-Port

ausgegeben und am Pin PB2 ein Strobe-Impuls erzeugt, der die Aussprache startet. All dies geschieht so schnell, daß man keine Verzögerung bemerkt.
Zum Schluß der Interrupt-Routine wird noch der oben schon erwähnte Zeiger aktualisiert. Er enthält immer die Adresse des nächsten Byte und muß wegen der Struktur der verwendeten LDA-Adressierungsart in der Zeropage stehen. Nachdem auch die Register auf ihren ursprünglichen Stand gebracht wurden, gibt die Interrupt-Routine mit RTI die Kontrolle wieder an das unterbrochene Hauptprogramm zurück, bis das Ende des Lautes wieder einen Interrupt auslöst.
Die eben beschriebenen Vorgänge bei der Interrupt-Abarbeitung gelten in ähnlicher Weise für den maskierbaren Interrupt IRQ. Der Vollständigkeit halber seien hier die Unterschiede kurz erwähnt: Die Startadresse der IRQ-Routine steht in den Speicherzellen $FFFE und $FFFF. Im Original-ROM ist sie $FF48. Hier werden zunächst alle Register gerettet und mit Hilfe des Break-Flag vom Stapel getestet, ob die Break-Routine angesprungen werden muß. Dies erfolgt dann über den RAM-Vektor $0316/$0317 (790/791, Normalinhalt $FE66), während die Interrupt-Routine über $0314/$0315 (788/789, Normalinhalt $EA31) erreichbar ist. Im Unterschied zur NMI-Routine sind also hier beim Sprung über die RAM-Vektoren bereits die Register auf den Stack gelegt.
Das Programm SC-01 INTERR.BAS enthält die gerade beschriebene Maschinenroutine in DATAs. Es kann mitsamt der Unterroutinen ab Zeile 10 000 in eigene Programme eingebaut werden. Die Zeilen 100 bis 190 stellen ein Beispielprogramm dar, das die Handhabung demonstrieren soll.
Das Unterprogramm ab Zeile 10 200 muß nur einmal zu Anfang des Programms aufgerufen werden. Es installiert zunächst die Maschinenroutine im Kassettenpuffer und ändert dann den NMI-Vektor im RAM auf die neue Routine. Danach werden noch der User-Port und das Port-Bit PB2 auf Ausgang programmiert, und das Flag-Bit des CIA # 2 als Interruptquelle festgelegt.
Zum Starten des Sprechvorgangs muß sich natürlich eine Phonemkette im Speicher befinden. Sie kann vorher mit dem SC-01 COMPILER dort abgelegt oder nachgeladen worden sein. Mit den zwei POKE-Befehlen in den Zeilen 160 und 170 wird die Anfangsadresse der Phonemkette festgelegt. Dann folgt der Aufruf des zweiten Unterprogramms, das die Übertragung anstößt, indem einfach der Phonemwert 3 (Pause) an den SC-01 ausgegeben wird. Der Rechner steht wieder für weitere BASIC-Befehle zur Verfügung, während im Hintergrund die Sprachausgabe läuft, bis in der Phonemkette der End-Kennwert $FF (255) erreicht wird.
Vom BASIC-Programm aus können Sie sogar überprüfen, wie weit die Ausgabe fortgeschritten ist, indem Sie die aktuelle Phonemadresse mit

```
A=256*PEEK(252)+PEEK(251)
```

auslesen. So können beispielsweise Aktionen gestartet werden, wenn die Sprachausgabe einen bestimmten Punkt überschritten hat.
Zum Testen laden Sie zuerst SC-01 INTERR.BAS und anschließend den Beispieltext TXT.DEMO. Mit RUN können

Sie nun die Hintergrundsprachausgabe starten. Der Cursor erscheint sofort wieder und der C64 steht auch während der Sprachausgabe wie gewohnt zu Ihrer Verfügung. Auch ein paralleler Diskettenzugriff ist möglich und beeinträchtigt die Sprachausgabe in keiner Weise.

2.4 Die Timer im CIA 6526

Außer Daten hin- und herzuschaufeln, kann der C64 noch mehr. Seine Stärken sind besonders beim Messen zu finden. Was die Timer dazu können, erfahren Sie jetzt.

2.4.1 Hardwarebasis

2.4.1.1 Zähler eingebaut

Aus der Zusammenfassung aller Funktionseinheiten des 6526 wissen Sie bereits, daß jeder der beiden CIA-Bausteine im C64 zwei 16-Bit-Timer enthält. Die Tabelle im Anhang B zeigt wieder kurz und prägnant die Programmierungsmöglichkeiten, die Sie im folgenden genauer anschauen wollen.

Da beide Bausteine völlig gleich aufgebaut sind und sich nur dadurch unterscheiden, daß ihre Register in verschiedenen Speicherbereichen ansprechbar sind soll hier statt der höherwertigen Adreßteile $DC und $DD einfach $XX geschrieben werden.

Die beiden Timer jeder CIA, Timer A und Timer B genannt, sind nichts anderes als Zähler, die grundsätzlich genauso aufgebaut sind, wie der einfache Flipflop-Zähler aus Kapitel 1.7.3. Jedoch sind im 6526 jeweils 16 Flipflops hintereinandergeschaltet. Somit kann ein einzelner Zähler von 1111111111111111 bis 0000000000000000 laufen, also 2^{16} = 65536 verschiedene Stellungen einnehmen. Wird eine Taktfrequenz am Zählereingang angelegt, dann schaltet er der Reihe nach alle Zustände von oben nach unten (!) durch.

Der jeweilige Zählerstand kann vom Computer gelesen werden. Da jeder der beiden Zähler 16 Bit breit ist, stehen jeweils zwei Register zur Verfügung. Es handelt sich um die Adressen $XX04 (also $DC04 bei CIA # 1 und $DD04 bei CIA # 2) bis $XX07. Die Zuordnung ist:

$XX04	Timer A Lowbyte
$XX05	Timer A Highbyte
$XX06	Timer B Lowbyte
$XX07	Timer B Highbyte

Diese Register legen bei Schreibzugriffen den Bereich fest, in dem der Zähler arbeiten soll.

Schön, werden Sie sagen, doch was nutzt einem das alles?

2.4.1.2 Gut gezählt, Computer

Die Zähler im 6526 sind über Steuerregister in vielfältiger Weise programmierbar und bieten eine Fülle von Anwendungsmöglichkeiten.

Das Register $XX0E kontrolliert den Timer A, $XX0F den Timer B. Im einzelnen haben die Bits in diesen Registern folgende Bedeutung:

Register $XX0E: (Kontrollregister Timer A)

Bit 0: 1=Timer A Start
0=Timer A Stop

Bit 1: 1=Unterlauf von Timer A wird an PB6 signalisiert
0=keine Signalisierung des Unterlaufs

Bit 2: falls Bit 1 gesetzt ist:
1=jeder Unterlauf von Timer A kippt PB6 in die jeweils andere Lage
0=jeder Unterlauf von Timer A erzeugt an PB6 einen High-Impuls mit der Länge eines Systemtaktes

Bit 3: 1=Timer A zählt nur einmal vom Ausgangswert auf 0, lädt den Zähler wieder mit dem Speicherinhalt der Latches $XX04 (Lowbyte) und $XX05 (Highbyte), und hält dann an
0=Timer B zählt fortlaufend über den im Latch befindlichen Ausgangswert auf 0

Bit 4: 1=unabhängig davon, ob der Timer gerade läuft oder nicht, wird das Zählregister mit den Werten aus $XX04 und $XX05 geladen
0=keine Funktion

Bit 5: 1=Timer A zählt steigende CNT-Flanken
0=Timer A zählt Systemtaktimpulse

Bit 6: 1=Serieller Port (SP) ist Ausgang
0=Serieller Port ist Eingang

Bit 7: Echtzeituhr läuft mit
1=50 Hz
0=60 Hz

Register $XX0F: (Kontrollregister Timer B)

Bit 0: 1=Timer B Start
0=Timer B Stop

Bit 1: 1=Unterlauf von Timer B wird an PB7 signalisiert
0=keine Signalisierung des Unterlaufs

Bit 2: falls Bit 1 gesetzt ist:
1=jeder Unterlauf von Timer B kippt PB7 in die jeweils andere Lage
0=jeder Unterlauf von Timer B erzeugt an PB7 einen High-Impuls mit der Länge eines Systemtaktes

Bit 3: 1=Timer B zählt nur einmal vom Ausgangswert auf 0, lädt den Zähler wieder mit dem Speicherinhalt von $XX06 (Lowbyte) und $XX07 (Highbyte) und hält dann an
0=Timer B zählt fortlaufend über den im Latch befindlichen Ausgangswert auf 0

Bit 4:	1=unabhängig davon, ob der Timer gerade läuft oder nicht, wird das Zählregister mit den Werten aus $XX06 und $XX07 geladen 0=keine Funktion
Bit 5, Bit 6:	Triggerquelle für Timer B: 00=Timer B zählt Systemtakte 10=Timer B zählt steigende CNT-Flanken 01=Timer B zählt Unterläufe von Timer A 11=Timer B zählt nur Unterläufe von Timer A, wenn CNT High ist
Bit 7:	Schreibzugriffe auf Register $XX08 bis $XX0B gehen ins 1=Alarmregister 0=Uhrzeitregister

Tabelle 2.21: Die Kontrollregister des CIA 6526

Im Unterschied zu dem Flipflop-Zähler aus Kapitel 1.7.3 lassen sich die Timer im CIA-Baustein auf einen bestimmten Startwert einstellen, sprich, sie lassen sich parallel laden. Um einen Zähler mit einem bestimmten Startwert loslaufen zu lassen, muß dieser Wert erst als High- und Lowbyte in die entsprechenden Register geschrieben werden. Danach setzt man Bit 4 im Kontrollregister des gewünschten Timer. Genau zu diesem Zeitpunkt übernimmt der Zähler den voreingestellten Wert.

2.4.1.3 Die Reaktion der Timer

Offensichtlich spielt der Zeitpunkt, zu dem der Zähler den Wert 0 erreicht, eine herausragende Rolle. Alle Aktionen kreisen um dieses Ereignis. In jedem Fall wird die Zählerkette dann auf den Wert voreingestellt, der durch Einschreiben in die Register $XX04 / $XX05 für Timer A bzw. $XX06 / $XX07 für Timer B zuletzt festgelegt wurde. Sofern Bit 3 gelöscht ist, geht der Zählvorgang anschließend normal weiter. War dieses Bit jedoch gesetzt, bleibt der Zähler auf dem programmierten Ausgangswert stehen.

Der Nulldurchgang wird auch als Unterlauf (englisch: Underflow) bezeichnet. Er kann auf verschiedene Weisen signalisiert werden.

Zunächst mittels der Portleitungen PB6 bei Timer A bzw. PB7 bei Timer B. Dazu muß Bit 1 gesetzt sein. Zu beachten ist, daß bei dieser Betriebsart die Portleitungen PB6 bzw. PB7 immer Ausgänge sind, unabhängig von der Einstellung im Datenrichtungsregister!

Sie werden sich vielleicht schon gefragt haben, woher die Timer ihren Takt bekommen. Mit Bit 5 im Kontrollregister A bzw. Bit 5 und 6 im Kontrollregister B sind verschiedene Quellen möglich.

Timer A kann nur vom Systemtakt oder aber von außen her angesteuert werden. Dagegen ist Timer B wesentlich flexibler. Er kann auch mit Unterläufen seines Kollegen Timer A getaktet werden, und das sogar wahlweise unter der Bedingung, daß der externe Anschluß CNT auf HIGH

liegt. Mit dieser Funktion ist es möglich, die beiden Zähler zu kaskadieren, also quasi zu einem 32-Bit-Zähler zusammenzukoppeln. Die maximale Stellung ist dann 4 294 967 294. Taktet man diesen zusammengeschalteten Zähler mit der Systemfrequenz von etwa 1 MHz, dann lassen sich Verzögerungszeiten von bis zu 71,6 Minuten erreichen.

Nützlich für Programmbeeinflussungen ist auch die Signalisierungsmöglichkeit eines Unterlaufs im Interrupt-Kontrollregister $XX0D, dessen Funktion im folgenden näher erläutert werden soll.

2.4.1.4 Bitte ruhig stören!

Die Bits 0 bis 4 im Interrupt-Kontrollregister zeigen den Zustand der verschiedenen Funktionseinheiten des CIA-Bausteins an. Sie werden gesetzt, wenn das zugehörige Ereignis stattgefunden hat. Durch Lesen dieses Registers werden jedoch alle Bits gelöscht!

Im einzelnen haben die Bits folgende Bedeutung:

Register $XX0D:	(Interrupt-Kontrollregister)
Bit 0:	Unterlauf Timer A
Bit 1:	Unterlauf Timer B
Bit 2:	Uhrzeit = Alarmzeit
Bit 3:	SDR voll/leer (je nach Datenrichtung)
Bit 4:	negative Flanke am Pin $\overline{\text{FLAG}}$ aufgetreten
Bit 5:	(Immer 0)
Bit 6:	(Immer 0)
Bit 7:	Übereinstimmung mindestens eines Bits mit der Interruptmaske

Tabelle 2.22: Das Interrupt-Kontrollregister

Die Interruptmaske ist eine Bitkombination, die durch Schreiben in dasselbe Register $XX0D festgelegt wird. 0-Bits lassen die entsprechenden Stellen des Interrupt-Kontrollregisters unbeeinflußt. Für die 1-Bits legt Bit 7 fest, ob gesetzt oder gelöscht werden soll. Ist Bit 7 gleich 0, löschen alle 1-Bits das entsprechende Maskenbit, ist Bit 7 dagegen 1, dann setzen sie es.

Falls ein Registerbit und ein Maskenbit gleichzeitig gesetzt sind, so wird auch das Registerbit 7 gesetzt und ein Interrupt ausgelöst. Diese Möglichkeit wird vom Betriebssystem des C64 ausgiebig genutzt, indem beide Zähler in CIA # 1 kaskadiert fortlaufend mit dem Systemtakt betrieben werden. So tritt in regelmäßigen Abständen ein Nulldurchgang auf, der einen Interrupt auslöst.

Damit wäre auch der Begriff Timer (Zeitgeber) geklärt.

2.4.1.5 Quarzstabil, aber auch quarzgenau?

In den nächsten Abschnitten sollen einige Meßgeräte vorgestellt werden, die eng mit den Timern zusammenarbeiten und sich auf den Systemtakt beziehen. Dieser Systemtakt ist zwar quarzgesteuert und damit in sich sehr stabil, die Commodore-Tech-

niker haben aber ihrem Oszillator nicht getraut und zusätzlich noch eine Einstellmöglichkeit in das System integriert. Vom Werk aus soll der Systemtakt damit hierzulande auf genau 985 248,444 Hz abgeglichen werden. Man kann sich denken, daß diese Einstellung nicht unbedingt mit großer Präzision vorgenommen wird. Demzufolge hat jeder C64 eine leicht unterschiedliche Betriebsfrequenz, was im normalen Einsatz keineswegs störend ist. Sollen dieser Frequenz aber Messungen zugrunde liegen, muß sie als Referenzgröße genau bekannt sein. Es ist daher ratsam, einmalig die Systemtaktfrequenz mit einem geeichten Frequenzzähler genau zu messen, beispielsweise bei einem Radio- und Fernsehtechniker. Die Meßgröße kann dabei – ohne Öffnen des Gehäuses – am Expansion-Port (rechts hinten am C64) abgegriffen werden. Zuständig sind die Anschlüsse A (Masse) und E (∅2).

Sollten Sie keine Möglichkeit für eine genaue Messung haben, hilft zur Not auch das Programm SYSTEMTAKTMETER weiter. Es geht von der Tatsache aus, daß die Netzfrequenz eine hervorragende Langzeitstabilität aufweist. Zu sehen ist das beispielsweise bei Radioweckern, die am Netz betrieben werden und auch nach langem Betrieb kaum Abweichungen zeigen.

Mit dieser Netzfrequenz werden auch die Hardware-Uhren in den CIA-Chips betrieben. Das Programm SYSTEMTAKTMETER mißt nun eine gewisse Zeit mit einer CIA-Uhr ab und zählt derweil die Taktimpulse. Anschließend errechnet es daraus die Taktfrequenz.

Trotz der Langzeitstabilität der Netzfrequenz sind kurzzeitige Schwankungen nicht ganz auszuschließen. Das Ergebnis der Messung wird also um so genauer, je länger die Messung durchgeführt wird. Am sichersten ist es, wenn mehrere Messungen zu unterschiedlichen Tageszeiten durchgeführt, und die Einzelergebnisse dann gemittelt werden.

In den Meßprogrammen der Zusätze befindet sich eine Stelle, an der die jeweilige Taktfrequenz des benutzten Einzelgerätes eingesetzt werden kann. Darauf wird aber zu gegebener Zeit noch hingewiesen.

2.4.2 Der C64 als Stoppuhr

2.4.2.1 Zeiten messen

Die Nutzung der Timer für externe Hardware geschieht vor allem über den Anschluß CNT. Diese Abkürzung steht für Count (= zählen). Derselbe Pin erfüllt gleichzeitig mehrere Aufgaben, je nach Programmierung, die aber immer mit dem Takt der Timer zu tun haben. Am User-Port sind die CNT-Pins beider CIAs zugänglich (siehe Anhang C). Da die Timer in CIA # 1 laufend zur Systeminterruptgenerierung verwendet werden, wollen wir zunächst nur CIA # 2 und seinen Anschluß CNT2 benutzen.

Mit den Steuerregistern können Sie die Timer auf diverse Betriebsarten programmieren. Am flexibelsten zeigt sich Timer B, für den die beiden Bits 5 und 6 im Kontrollregister B verantwortlich zeichnen. Ihre Funktion wurde bereits erläutert. Hier soll die Kombination Bit 5 = 1 / Bit 6 = 1 eingesetzt werden: Timer B zählt nur dann, wenn der Pin CNT2 HIGH führt, bei LOW bleibt er dagegen unverändert. Die Geschwindigkeit, mit der erzählt, wird durch Timer A festgelegt. Je-

desmal, wenn der durch 0 geht und wieder mit seinem Startwert geladen wird, vermindert Timer A seinen Zählerinhalt einmal. Wird Timer A mit dem Systemtakt des C64 betrieben, dann kennt man genau die Unterlaufzeit und erhält bereits eine Stoppuhr! Sie könnten einen Taster anschließen, zum Beispiel den aus Kapitel 1.5, und genau messen, wie lange er gedrückt bleibt, indem Sie die Zähler vorher auf ihren Maximalwert stellen und nach der Aktion den Inhalt auswerten.

2.4.2.2 Eine vielseitige Meßplatine

Die vorgeschlagene Methode ist zugegebenermaßen etwas umständlich, und über ihren Nutzen läßt sich streiten. Deshalb wurden eine kleine Zusatzschaltung und ein Programm entwickelt, das Messungen mit unterschiedlichen Signalgebern ermöglicht. So können beispielsweise Lichtschranken angeschlossen werden, mit denen exakte Zeit- und auch Geschwindigkeitsmessungen durchführbar sind.
Die Schaltung des Zusatzes zeigt Bild 2.53.

Auf den ersten Blick fällt die Ähnlichkeit mit der Elektronik des entprellten Tasters aus Kapitel 1.5 ins Auge. Auch hier handelt es sich um ein einfaches Setz-Rücksetz-Flipflop, das zusätzlich über einen Steuereingang verfügt. Zunächst soll dieser vom User-Port-Anschluß PA2 bediente Steuereingang HIGH sein. Die Funktion entspricht in diesem Fall genau der des Tasters und wurde dort bereits detailliert erläutert. An dieser Stelle nur soviel: Die Eingänge reagieren auf LOW-Impulse. Normalerweise sind sie HIGH. Wird $\overline{E_1}$ kurz LOW, geht der Ausgang – also die Verbindung zu CNT2 und $\overline{\text{FLAG2}}$ – sofort nach HIGH und bleibt dort. Dieser Zustand kann geändert werden, indem $\overline{E_2}$ kurz LOW gemacht wird. In diesem Fall kippt das Flipflop in seinen anderen Zustand. Der Ausgang wird also wieder LOW und bleibt auch LOW, bis wieder ein Signal an $\overline{E_1}$ auftritt.
Das dritte Gatter zeigt den Zustand des Flipflop über eine Leuchtdiode an. Leuchtet sie, dann ist der Ausgang HIGH, das heißt, der angeschlossene, entspre-

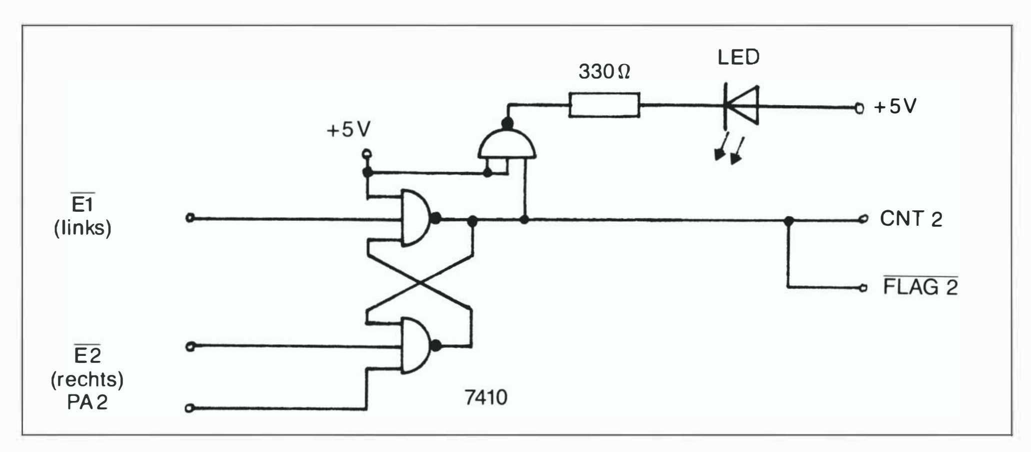


Bild 2.53: Die Schaltung zur Stoppuhr »Zähler«

chend programmierte Zähler im Computer würde laufen, ist sie dagegen dunkel, ist der Flipflop-Ausgang LOW.

Der mit PA2 verbundene Steuereingang hat zwei Funktionen. Einmal löscht er mit einem kurzen LOW-Impuls das Flipflop, zum anderen läßt sich über durchgehenden LOW-Pegel an diesem Pin die Speicherwirkung des Flipflop abschalten, so daß $\overline{E_1}$ nun direkt auf den Ausgang wirkt. $\overline{E_2}$ bleibt in diesem Fall wirkungslos. Für den Betrieb als Zeitgeber eröffnen sich damit diverse Meßmöglichkeiten.

Der Ausgang des Flipflop wurde außer auf CNT2 auch noch auf $\overline{\text{FLAG2}}$ geschaltet. So kann das Ende einer Messung leicht von der Software erkannt werden, denn zu diesem Zeitpunkt geht ja der Flipflop-Ausgang von HIGH nach LOW, das heißt, die fallende Flanke setzt das Flag-Bit im Interrupt-Kontrollregister.

2.4.2.3 Aufbau der Platine »Zähler«

Die Zusatzschaltung besteht nur aus wenigen Bauteilen, die auf der Platine nach Bild 2.54 mit Hilfe des Bestückungsplans (Bild 2.55) schnell verlötet sind. Tabelle 2.23 enhält alle benötigten Teile.

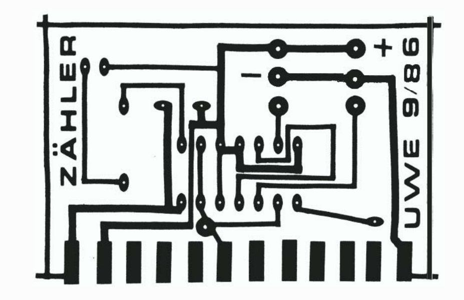


Bild 2.54: Layout für eine vielseitige Platine zur Ansteuerung der Zahlen

1	Logik-IC 74LS10
1	Sockel 14polig
1	Leuchtdiode, Farbe nach Wahl
1	Widerstand 330 Ohm
1	Kondensator 0,1 Mikrofarad, Keramik
6	Lötstifte
1	User-Port-Stecker
1	einseitige Platine nach Bild 2.54

Tabelle 2.23: Einzelteile der Platine »Zähler«

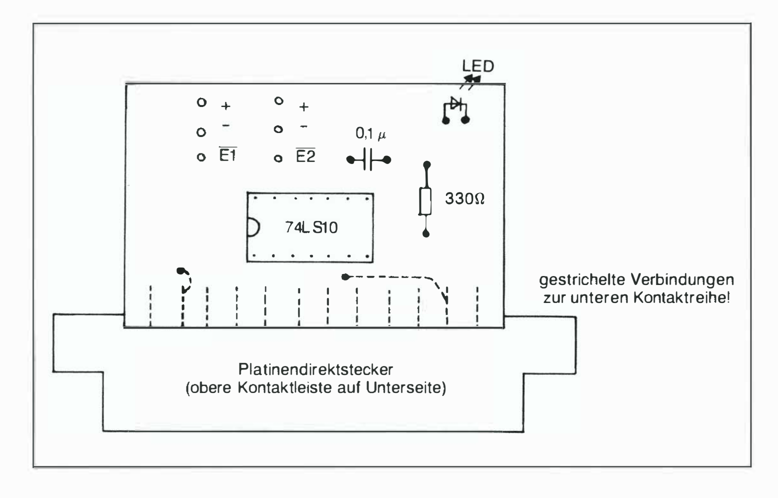


Bild 2.55: Bestückungsplan zur »Zähler«-Platine

Das IC 74LS10 beherbergt alle drei Gatter. Es sollte auf einen Sockel gesetzt werden. Wichtig ist die Anbringung des User-Port-Steckers. Die Platine wird auf seine obere (!) Kontaktreihe aufgelötet und die beiden gekennzeichneten Pins der unteren Reihe mit den angegebenen Lötaugen verbunden. Achten Sie auf die Polarität der LED und die Einbaurichtung des IC. Weitere Hinweise befinden sich im Anhang A.

Foto 2.8 zeigt die Stoppuhr zusammen mit dem Lichtschrankenmodul aus Kapitel 2.2.4.4.

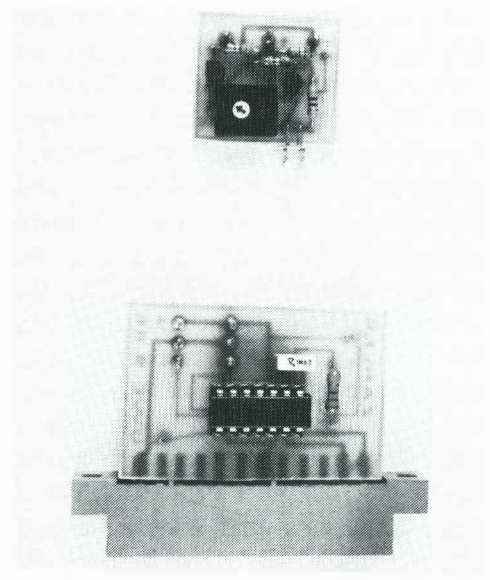


Foto 2.8: Zähler-Platine und Lichtschrankenmodul als Signalgeber

2.4.2.4 Betrieb der Stoppuhr

Nun kann's losgehen. Als Signalquelle eignen sich Lichtschrankenmodule aus Kapitel 2.2.4.4. Schließen Sie ein Modul an die drei Pins von $\overline{E_1}$ an, das andere bei $\overline{E_2}$ (Polarität beachten!). Dann stecken Sie die Zusatzplatine auf den User-Port und schalten den Rechner ein. Laden Sie das BASIC-Programm STOPPUHR und starten es mit RUN.

Antworten Sie auf die Frage nach der Betriebsart zunächst mit 1. In diesem Fall legt der Computer LOW an den Steueranschluß PA2, und es ist nur die Lichtschranke an $\overline{E_1}$ aktiv. Die Leuchtdiode zeigt direkt ihren Zustand an: Ist der Strahl unterbrochen, leuchtet sie auf, andernfalls bleibt sie dunkel.

Als nächstes folgt die Eingabe des Meßbereiches. Je kleiner die maximale Meßzeit ist, um so genauer kann anschließend das Ergebnis ermittelt werden. Wählen Sie zunächst mit 3 den Bereich bis 10 Sekunden.

Auf die Frage nach dem Anzeigemodus antworten Sie mit 1 und nach der Geschwindigkeitsmessung mit N.

Es erscheint das Meßbild. Sobald die Lichtschranke unterbrochen wird, läuft die Anzeige los und stoppt, wenn die Unterbrechung endet. Mit W können Sie die Messung wiederholen.

Wählen Sie nun mit N die Einstellung der Betriebsart wieder an. Antworten Sie wie vorher, aber geben Sie bei der zweitletzten Frage 2 für automatische Meßwiederholung mit ausschließlicher Anzeige des jeweiligen Endwertes an.

Nun verfahren Sie wie vorher. Sie müssen nicht jedesmal mit W eine neue Messung einleiten, sondern nach jeder Unterbrechung wird sofort deren Dauer angezeigt, bis die folgende Messung beendet ist. Diese Funktion ist besonders zur Beobachtung sich ständig wiederholender Abläufe geeignet. Nehmen Sie beispielsweise ein Pendel, das bei jeder Schwingung die Lichtschranke kurz unterbricht.

Aus der Dauer der Unterbrechung kann

leicht die Geschwindigkeit des Objektes, das die Verdeckung verursacht, berechnet werden, wenn seine Länge bekannt ist. Auch solche Geschwindigkeiten kann die Stoppuhr-Software anzeigen. Unterbrechen Sie dazu die laufende Messung mit N und beantworten Sie die entsprechende Frage jetzt mit J. Sie müssen zusätzlich noch die Länge des Objektes in Zentimetern angeben.
Nach jeder Messung wird in der unteren Bildschirmhälfte seine Geschwindigkeit in Metern pro Sekunde (m/s) und in Stundenkilometern (km/h) angezeigt.
Setzt man einen Inverter zwischen Lichtschrankenmodul und Zählerplatine, dann lassen sich auch Hellzeiten messen, beispielsweise die Leuchtzeit eines Blitzes.
Das Gerät eignet sich damit besonders für Fotofreunde, denn somit läßt sich die Belichtungszeit eines Blitzgerätes messen. Ebenso läßt sich der Modul hinter einem geöffneten Fotoapparat anbringen, während die Linse von vorne gut beleuchtet wird. Drückt man auf den Auslöser, dann zeigt der C 64 die genaue Verschlußzeit an.

Der Betrieb mit zwei Signalgebern – in unserem Falle also mit zwei Lichtschranken – läuft ähnlich. Hier wird jedoch die Messung bei einem LOW-Impuls am Eingang $\overline{E_1}$ gestartet und bei einem LOW-Impuls an $\overline{E_2}$ beendet. Wieder ist Einzelmessung und automatische Wiederholung mit Anzeige der Endwerte möglich. Auch die Geschwindigkeit kann ermittelt werden, wobei jedoch nun der Abstand beider Signalgeber voneinander maßgeblich ist. Die Länge des Meßobjekts spielt keine Rolle.
Mit dieser Funktion könnte zum Beispiel die Geschwindigkeit der Rennwagen auf Modellauto-Rennbahnen getestet werden. Eine andere Möglichkeit, über die Zeit Meßgrößen zu erfassen, zeigt der folgende Abschnitt.

2.4.3 Kondensator- und Widerstandsmeßgerät

2.4.3.1 Monoflop als Zeitgeber

Jeder Elektronikbastler kennt das Problem: In der Bauteilkiste liegen etliche undefinierbare Kondensatoren mit Aufschriften, wie zum Beispiel 22000F oder 20j. Woher soll mit Sicherheit festgestellt werden, ob das herausgefischte Exemplar auch die richtige Zehnerpotenz besitzt? Zwar geben schon die Bauteilabmessungen einen Anhaltspunkt, ob Pikofarad, Nanofarad oder gar Mikrofarad gemeint sind, die zulässige Spannung bestimmt mitunter die Größe. So entsteht der Wunsch, die Bauteile nachmessen zu können, doch Kapazitätsmeßgeräte sind nicht gerade billig, und eine Anschaffung lohnt sich in den meisten Fällen nicht.
Dank der universell programmierbaren Zähler läßt sich aber der C 64 leicht zu einem Kapazitäts- und Widerstandsmeßgerät aufrüsten, das einen Vergleich mit professionellen Systemen nicht zu scheuen braucht.
Das Funktionsprinzip ist ganz einfach: Ein ganz normales Monoflop aus der TTL-Familie dient als Zeitgeber. Erinnern Sie sich: Ein Monoflop reagiert auf einen LOW-HIGH-Sprung an seinem Eingang mit einem Ausgangsimpuls, dessen Dauer über eine RC-(Widerstands-Kondensator-)Kombination einstellbar ist (Kapitel 2.2.4.6). Verändert man den Konden-

satorwert, dann ändert sich auch die Impulsdauer. Sie errechnet sich beim 74121 nach folgender Formel:

$$T = R * C * \ln(2)$$

Auch die HIGH-Zeit des Monoflop-Ausgangs kann mit den Timern gemessen werden, indem man in Register 15 Bit 5 und Bit 6 setzt. Wie Sie aus dem vorigen Abschnitt wissen, wird das Monoflop dazu an den Pin CNT einer CIA angeschlossen. Da die Timer in CIA # 1 bereits vom Betriebssystem benutzt werden, tritt jetzt CIA # 2 in Aktion. Timer B zählt jetzt nur, solange CNT2 HIGH ist. Wird Timer A mit der bekannten Taktfrequenz des C64 angesteuert, dann ergibt sich aus der Anzahl der gezählten Impulse, multipliziert mit der Taktdauer (1/f) direkt die gemessene Zeit. Je nachdem, ob der Widerstand oder der Kondensator bekannt ist, läßt sich der fehlende Wert berechnen durch

$$C = \frac{T}{R * \ln(2)} \quad \text{oder} \quad R = \frac{T}{C * \ln(2)}$$

Dabei wird T in Sekunden, R in Ohm und C in Farad eingesetzt. Für ln(2) kann der gerundete Wert 0,7 benutzt werden.

2.4.3.2 Funktion durch Händeschütteln

Bild 2.56 zeigt die kleine Meßschaltung. Durch einen zweipoligen Umschalter ist es möglich, zwischen Kondensator- und Widerstandsmessung zu wählen, indem einmal ein bekannter Widerstandswert, im anderen Fall aber ein fester Kondensator eingestellt werden kann.

Wieder wurden die Handshake-Leitungen zur Steuerung der Aktionen benutzt. Durch einen Schreib- oder Lesebefehl auf die Portadresse 56 577 wird der Meßimpuls ausgelöst, weil ja danach $\overline{PC2}$ automatisch für die Dauer eines Taktimpulses auf LOW geht. Die positive Flanke trig-

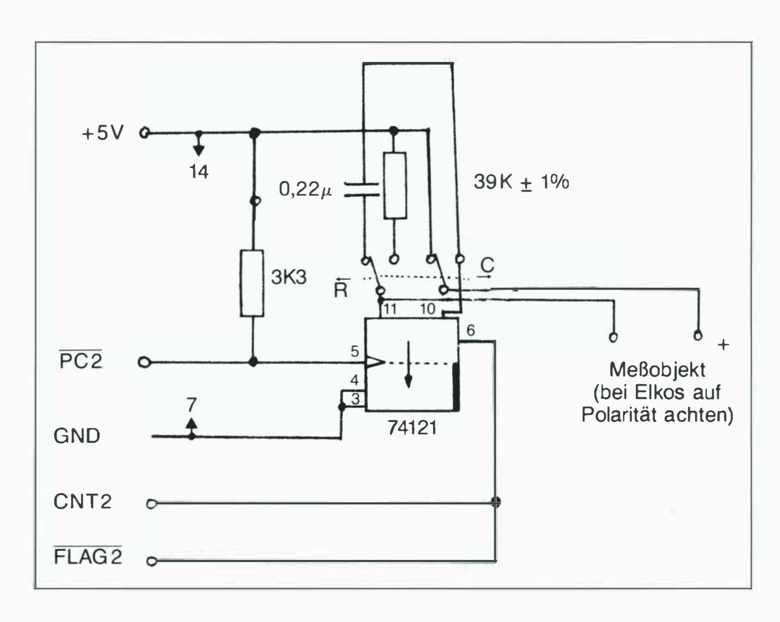


Bild 2.56: Schaltung des Kapazitäts- und Widerstandsmeßgerätes

gert das Monoflop. Beim zugehörigen Programm C/R-MESSGERAET geschieht das in Zeile 3300.
Aus Tabelle 2.21 ist die Programmierung der Timer ersichtlich.
Timer A wird zur Messung mit einem bestimmten Wert geladen und zählt von dort aus abwärts. Erreicht er die Stellung 0, dann wird selbsttätig der Inhalt von Timer B um 1 erniedrigt, Timer A wird erneut mit dem vorgegebenen Startwert geladen, und der Vorgang wiederholt sich so lange, bis das Monoflop wieder in seine Ruhelage zurückkehrt.
Der Monoflop-Ausgang Q ist außer mit CNT2 gleichzeitig noch mit dem Eingang $\overline{\text{FLAG2}}$ des C64 verbunden. Die fallende Flanke dort setzt Bit 4 im Interrupt-Kontrollregister (Register 13: $DD0D = 56 589), das vom Programm in einer Warteschleife abgefragt werden kann. Ist es gesetzt, signalisiert es somit das Ende der Meßzeit und die Gültigkeit des Timerinhalts. Beim Auslesen wird dieses Bit selbständig wieder gelöscht. (Siehe Kapitel 2.3.1.)

2.4.3.3 Errechnung des Anzeigewertes und Bereichsumschaltung

Das Programm liest nach der eigentlichen Messung den Timer aus (2 Byte!), multipliziert den Inhalt mit der Taktzeit und berechnet mit einer der Formeln aus Abschnitt 2.4.3.1 den gesuchten Wert. Für den Bereich Pikofarad und Nanofarad (B=0) erfolgt diese Rechnung in Zeile 5020. Sollte die zu messende Zeit die Zählerkapazität überschreiten, so steht nach der Messung der Wert 0 in der Variablen und es wird in Zeile 5030 automatisch auf den Meßbereich Mikrofarad (B=1) umgeschaltet. Hierbei wird das Highbyte des Timers A (H) in Zeile 3210 nun mit 5 geladen, so daß längere Zeiten gemessen werden können. Die Umrechnung erfolgt jetzt in Zeile 6000. Falls die gemessene Zeit einen bestimmten Grenzwert unterschreitet, schaltet das Programm in Zeile 5035 wieder zurück zum Bereich 0.
Um genaue Ergebnisse zu gewährleisten, muß sichergestellt sein, daß nach einer Messung immer eine Erholzeit verstreicht, die mindestens 75% der Impulsdauer betragen soll. Diese Forderung ist normalerweise bereits durch die Programmlaufzeit erfüllt. Lediglich im Bereich Mikrofarad können längere Meßzeiten auftreten, die bei großen Kapazitäten schnell wachsen und nicht zu unterschätzen sind. Bei 1000 Mikrofarad beträgt die Impulsdauer bereits 27,3 Sekunden. Daher wird in Zeile 6020 eine angepaßte Warteschleife durchlaufen, die auch die Menüfunktionen zugänglich macht.

2.4.3.4 Aufbau und Inbetriebnahme

Die kleine Zusatzschaltung kann auch von weniger Geübten leicht aufgebaut werden. Alle Teile (Tabelle 2.24) sind handelsüblich und der gesamte Kostenaufwand liegt bei nur etwa DM 10,-.

1	Monoflop 74121
1	Widerstand 3,3 Kiloohm
1	Meßwiderstand 39 Kiloohm ± 1%
1	Kondensator 0,22 Mikrofarad
1	Schalter 2 x um, lötbar
1	User-Port-Stecker
1	einseitige Platine nach Bild 2.57

Tabelle 2.24 Die Bauteile des C/R-Meters

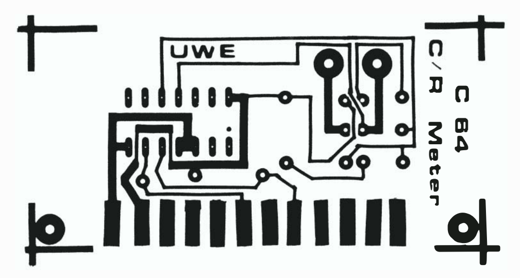


Bild 2.57: Layout für das C/R-Meßgerät

bei Schalterstellung »C« und Funktion »Kondensatormeßgerät« die mit »-« bezeichnete Meßleitung mit demjenigen Anschluß des Kondensators verbunden wird, der am Umschalter liegt. Der angezeigte Wert wird in Zeile 7050 eingetragen. War die Anzeige beispielsweise 211 Nanofarad, dann muß die Zeile lauten:

```
7050 W=(65535-(PEEK(56582)+
PEEK(56583)*256))*T/(.7*211E-9)
```

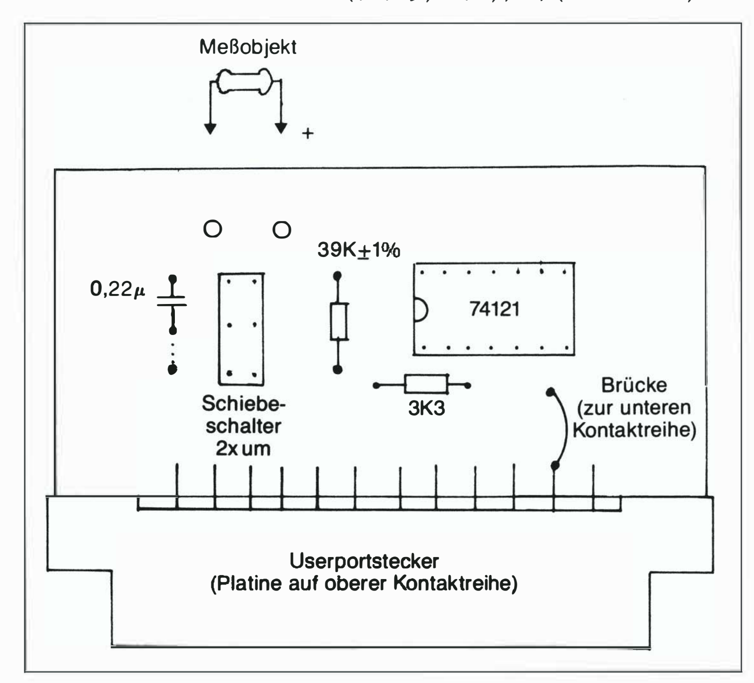


Bild 2.58: Bestückung der C/R-Meter-Platine

Bild 2.58 zeigt den Bestückungsplan der Meßplatine. Richten Sie sich beim Aufbau nach den Hinweisen in Anhang A. Um den Anzeigefehler möglichst klein zu halten, sollte für den 39-Kiloohm-Festwiderstand eine Ausführung mit kleiner Toleranz verwendet werden (± 1% oder kleiner).

Der Festkondensator für die Widerstandsmessung kann gemessen werden, indem

In Zeile 500 steht die Taktzeit des C64. Um genaue Ergebnisse zu erhalten, sollten Sie die Frequenz Ihres persönlichen Exemplars nach der Beschreibung aus Kapitel 2.4.1.5 ermitteln und den Kehrwert ebenfalls eintragen. Das Meßgerät ist nun geeicht und betriebsfertig.

Beim Start des Programms wird zuerst das Kapazitätsmeßgerät aktiviert. Dazu muß der Schalter auf der Zusatzplatine in Stel-

lung »C« gebracht werden. Auf der linken Bildschirmhälfte erscheinen zehn numerierte Speicherplätze, rechts leuchtet der Meßwert, sobald ein Objekt angeschlossen wird. Um die Benutzung zu erleichtern, erscheinen am unteren Bildrand alle verfügbaren Funktionen. Durch Drücken des Anfangsbuchstabens können sie angewählt werden.

Mit »S«, gefolgt von einer Ziffer, wird der angezeigte Wert in den entsprechenden Speicher übernommen.

»L« löscht alle gespeicherten Einträge,

»W« oder »R« schalten um auf Widerstandsmessung,

»K« oder »C« wieder zurück zur Kapazitätsmessung.

Gelegentlich kann es beim Wechseln von Meßobjekten vorkommen, daß das Gerät nicht mehr reagiert. Hier hilft es, mit »I« einen Impuls von Hand auszulösen.

Mit »Q« kann das Programm verlassen werden.

Der Meßbereich des Systems reicht bei der angegebenen Dimensionierung von der Größenordnung der Kabelkapazität bis zu einigen tausend Mikrofarad. Im Ohm-Meßbereich überdeckt es jedoch nur die Werte von etwa 100 Ohm bis ungefähr 150 Kiloohm.

2.4.4 Frequenzmeter mit dem C64

2.4.4.1 Features

Im Hobbylabor steht man oft vor dem Problem, Frequenzen oder Schwingungsdauern messen zu müssen. Ohne entsprechende Geräte ist das eine schier unlösbare Aufgabe. Wer in der glücklichen Lage ist, ein Oszilloskop sein eigen nennen zu dürfen, kann über die Kurvenform des Signals ungefähre Werte ermitteln, aber diese Methode ist im allgemeinen nicht sehr genau, und bei niedrigen Frequenzen ab etwa 10 Hz versagt sie ganz.

Mit äußerst geringem Aufwand an Zusatzhardware erhalten Sie hier ein Meßgerät, das seinesgleichen sucht. Wegen der Möglichkeit, mit extrem unterschiedlichen Torzeiten zu arbeiten, übertrifft es bei sorgfältiger Einstellung selbst teure Profigeräte an Genauigkeit. Die Anzeige umfaßt – dank des gegenüber LED-Anzeigen vergleichsweise sehr großflächigen Bildschirmdisplay – neben der Frequenz auch noch simultan die Angabe der Periodendauer, Daten zur Toleranz der Meßwerte und als Clou eine Mittelwertanzeige. Außerdem eröffnet sich die Möglichkeit zur programmgesteuerten Datenerfassung. Alles zusammen Eigenschaften, die bisher im Hobbybereich als unbezahlbar galten.

2.4.4.2 Wie ein Frequenzmeßgerät arbeitet

Gehen wir einmal von der Definition der Maßeinheit für Frequenzen aus. Sie ist nach dem Physiker Heinrich Hertz benannt, der 1888 als erster elektrische Wellen erzeugte. Ein Hertz (Abkürzung Hz) bedeutet eine Schwingung in jeder Sekunde. Sollen Frequenzen gemessen werden, sind also die Schwingungen eine Sekunde lang zu zählen. Nach diesem Prinzip arbeiten alle Frequenzmeßgeräte; sie werden daher auch Frequenzzähler genannt. Es ist ebenso möglich, über eine längere oder kürzere Zeit zu zählen, und dann das Ergebnis auf eine Sekunde umzurechnen.

Ein Frequenzzähler besteht damit eigent-

lich nur aus einer Einheit, die eine genau festgelegte Zeit lang die zu messende Frequenz auf einen Zähler wirken läßt, und dann das Zählergebnis auf einer Anzeige sichtbar macht. Das sollte der C64 doch auch können, zumal hier bereits zwei 32-Bit-Zähler eingebaut sind.

Das Prinzip ist leicht erklärt: Einer der CIA-Bausteine wird mit dem Systemtakt betrieben und erzeugt am User-Port die Meßzeit, einen HIGH-Impuls von genau einer Sekunde Länge. Der andere Zähler wird von außen mit der Meßfrequenz angesteuert. Damit er nur während der Meßzeit reagiert, muß die Meßfrequenz extern noch über ein UND-Gatter mit dem HIGH-Impuls verknüpft werden. Das Gatter wirkt dabei wie ein Tor, das geöffnet wird und die Meßfrequenz passieren läßt, falls der Impulseingang HIGH ist. Man nennt daher seine HIGH-Zeit auch Torzeit.

Dieses UND-Gatter ist das einzige unbedingt benötigte externe Bauelement. So kommt man als C64-Besitzer äußerst preisgünstig zu einem hochwertigen Meßgerät. Wie sieht es aber mit der Genauigkeit der Messung aus?

2.4.4.3 Wer mißt, mißt Mist

Dieser alte Spruch erweist sich leider immer wieder als richtig. Jede Messung ist in irgendeiner Weise fehlerbehaftet. Grundsätzlich kann zwischen zwei ganz verschiedenen Fehlerquellen unterschieden werden. Da gibt es auf der einen Seite die »Zufälligen« Fehler, die aufgrund von Unzulänglichkeiten der Bezugsgrößen auftreten, und auf der anderen Seite die »Systematischen« Fehler.

Fangen wir mit der ersten Gruppe an. Bei genauem Hinsehen hängt unser Meßergebnis eigentlich nur von der Genauigkeit der Zeitbasis und damit der Torzeit ab. Im C64 erzeugt ein Quarzoszillator die Taktfrequenz, mit der die Timer in den CIAs betrieben werden. Bei ungefähr 980 kHz erhält man eine Abstufung der einstellbaren Torzeit von $dt = 0{,}000001$ Sekunden. Daraus ergibt sich ein maximaler Fehler von

$$dt/t = 0{,}000001 = 0{,}0001\%$$

für eine Torzeit t von einer Sekunde: ein äußerst geringer Wert.

Zu dieser Abweichung addiert sich noch der »Systematische« Fehler, der – wie wir noch sehen werden – sehr viel größer sein kann.

Digitale Abläufe haben einen großen Vorteil: Neben den einzelnen diskreten Zuständen existieren keine Zwischenwerte, so daß man zu jeder Zeit ohne Ablesefehler genau sagen kann, welche Größe dargestellt wird. Es gibt nur Entweder-Oder. Gleichzeitig ist das aber ein großer Nachteil. Soll beispielsweise eine bestimmte lange Kommazahl mit einer festen Anzahl von Bits dargestellt werden, dann ist man angewiesen auf die nächste darstellbare Ziffernfolge, was immer zu Fehlern führt.

Das eben Gesagte trifft bei Meßfrequenzen zu, deren Schwingungsdauern nicht genau Vielfache der Torzeit sind. Ein Zähler registriert nur die positiven Flanken, also nur die LOW-HIGH-Übergänge der Meßfrequenz. Bild 2.59 zeigt die Verhältnisse für 5 Hz. Hier geht alles glatt. In einer Sekunde werden genau 5 Übergänge gezählt. Was aber passiert, wenn die Torzeit nicht genau eine Sekunde beträgt. Ist

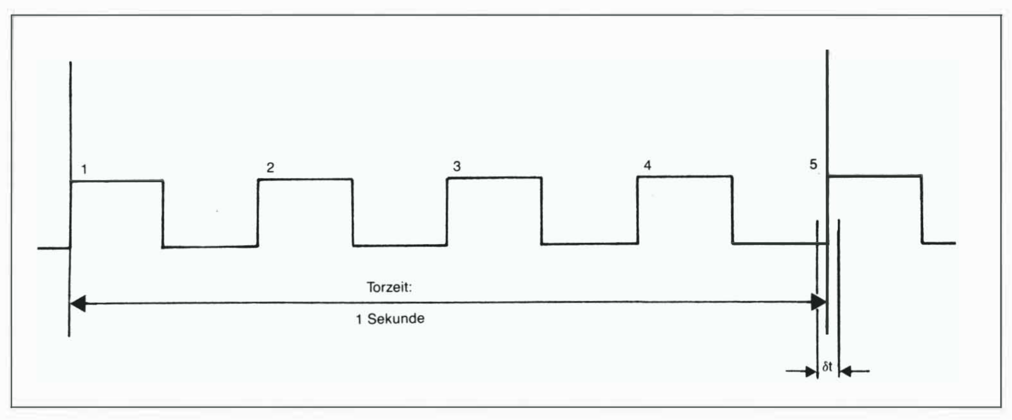


Bild 2.59: Einfluß der Torzeitschwankung auf das Meßergebnis

sie etwas länger, hat das keinen Einfluß auf das Zählergebnis. Sobald sie jedoch kürzer wird, bekommt die Schaltung die fünfte positive Flanke nicht mehr mit, und das Ergebnis wird falsch.
Eine andere Unsicherheit ist die Phasenlage der Meßfrequenz. In Bild 2.60 sind ebenfalls die Abläufe für etwa 5 Hz dargestellt. Die Schwingungen in Teil a) und b) haben exakt die gleiche Frequenz; sie sind lediglich etwas gegeneinander verschoben. Trotzdem werden bei a) 5 Impulse gezählt, im Fall b) jedoch nur 4.
Ganz allgemein kann man sagen, die Abweichung fm des Zählergebnisses Z vom tatsächlichen Wert beträgt maximal einen Zählimpuls, also

$$f_m = \frac{1}{Z} .$$

In unserem Beispiel wurden fünf Impulse gezählt. Die Meßfrequenz kann also zwischen 4 Hz und 6 Hz liegen. Das ergibt eine maximale Abweichung von

$$f_m = 1/5 = 0{,}2 = 20\% .$$

Wie zu errechnen, wird der Fehler aber

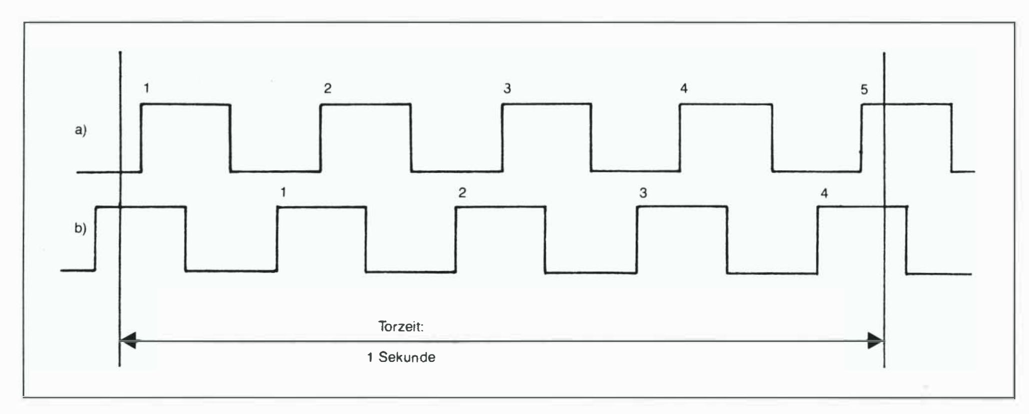


Bild 2.60: Einfluß der Phasenlage auf das Meßergebnis

mit wachsender Meßfrequenz schnell kleiner. Bei 100 kHz beträgt er nur noch maximal

$$f_m = 1/100000 = 0{,}00001 = 0{,}001\%\ .$$

Eine Möglichkeit, auch bei kleinen zu messenden Frequenzen die Genauigkeit zu steigern, eröffnet die Erhöhung der Meßzeit. Werden die aufgeführten 5 Hz beispielsweise mit einer Torzeit von 32 Sekunden gemessen, dann ist das Zählergebnis

$$Z = 5 * 32 = 160\ ,$$

und damit der maximale Fehler nur noch

$$f_m = 1/160 = 0{,}00625 = 0{,}625\%\ .$$

Für diese Ungenauigkeit ist das Meßverfahren verantwortlich. Sie tritt in genau derselben Größe auch bei professionellen Systemen auf. Mit diesen Geräten können aber oft längere Torzeiten als eine Sekunde nicht erzeugt werden, so daß unser C64-Frequenzmeter sogar im Vorteil liegt.

2.4.4.4 Aufbau des Frequenzmeters

Bild 2.61 zeigt den Schaltplan der Erweiterung. Obwohl eigentlich nur ein einziges Gatter benötigt wird, besteht die Stückliste (Tabelle 2.25) doch aus einigen Teilen mehr. Auf der Platine des Meßzusatzes (Bild 2.62) befindet sich zusätzlich ein re-

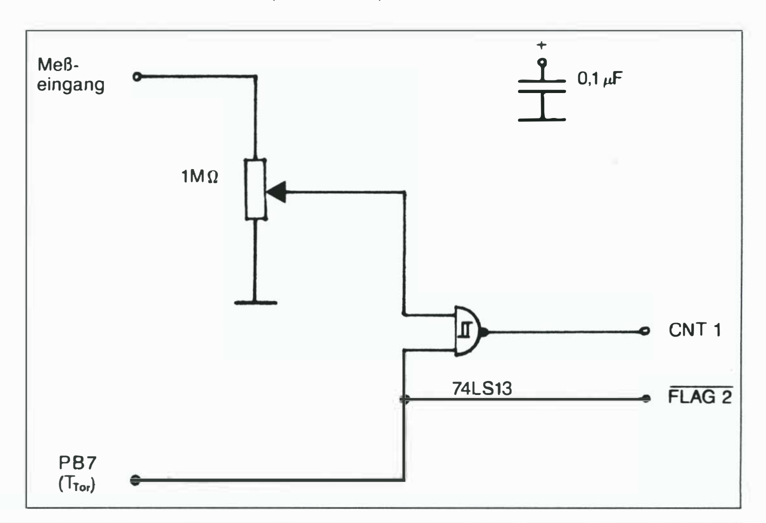


Bild 2.61: Schaltplan des Meßzusatzes

Bauteile	
1	74LS13 oder ähnliche (siehe Text)
1	IC-Sockel, 14polige
1	Trimmer 1M Ω, liegend
1	Kondensator 0,1 µF, Keramik
2	Lötnägel
1	Platine nach Bild 2.62
1	User-Port-Platinen-Direktstecker z.B. 251-12-50-170 von TRW

Tabelle 2.25: Der Einkaufszettel zum Frequenzmeter

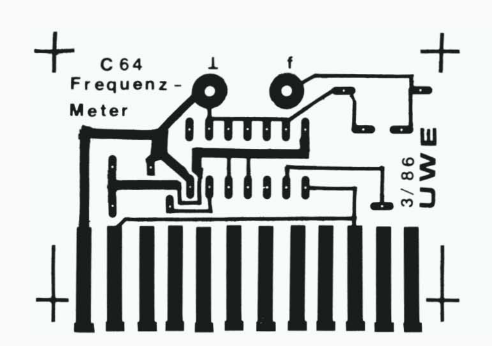


Bild 2.62: Layout zum Frequenzmeter

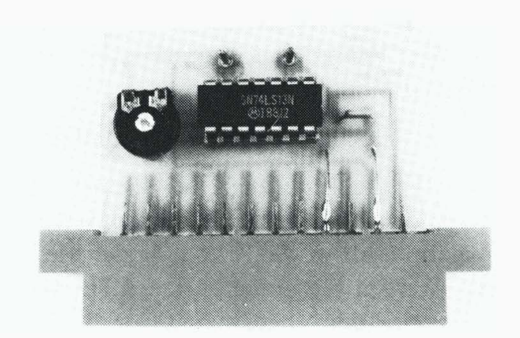
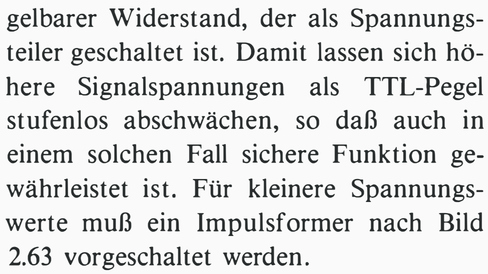


Foto 2.9: Die fertige Platine



gelbarer Widerstand, der als Spannungsteiler geschaltet ist. Damit lassen sich höhere Signalspannungen als TTL-Pegel stufenlos abschwächen, so daß auch in einem solchen Fall sichere Funktion gewährleistet ist. Für kleinere Spannungswerte muß ein Impulsformer nach Bild 2.63 vorgeschaltet werden.

Die Bestückung der kleinen Platine (Foto 2.9) geschieht wieder nach dem Bestückungsplan (Bild 2.64) unter Beachtung der Hinweise aus Anhang A.

2.4.4.5 Das Meßprogramm

Wie bereits erwähnt, benötigt das Programm beide 32-Bit-Timer. Da das Betriebssystem normalerweise die Timer in CIA #1 als Interruptquelle verwendet, mußte die eigentliche Meßroutine in Assembler geschrieben werden. Der Quelltext FREQUMETER.ASM besteht fast ausschließlich aus LDA- und STA-Befehlen und dürfte daher auch für Maschinensprache-Anfänger gut verständlich sein. Er wurde zur Generierung der DATA-Zeilen im BASIC-Teil benutzt.

2.4.4.5.1 Der Assemblerteil des Meßprogramms

Mit den ersten Befehlen wird der normale Interruptbetrieb unterbunden. Dies geschieht durch Löschen der Interruptmaske

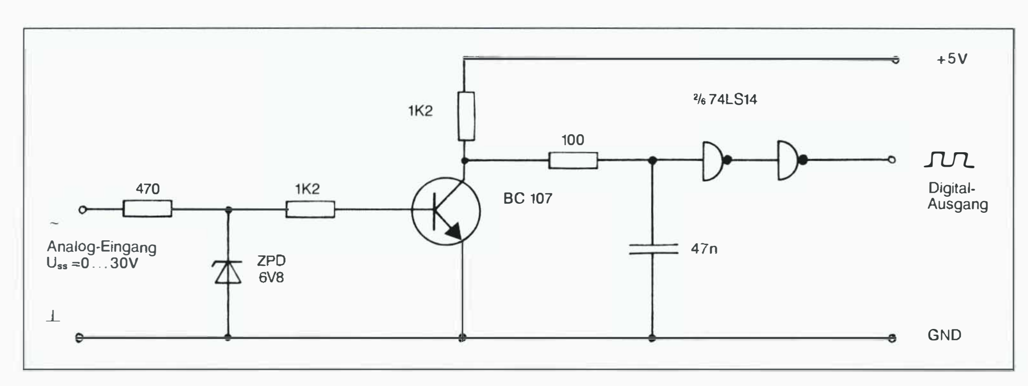


Bild 2.63: Impulsformer für den Frequenzzähler

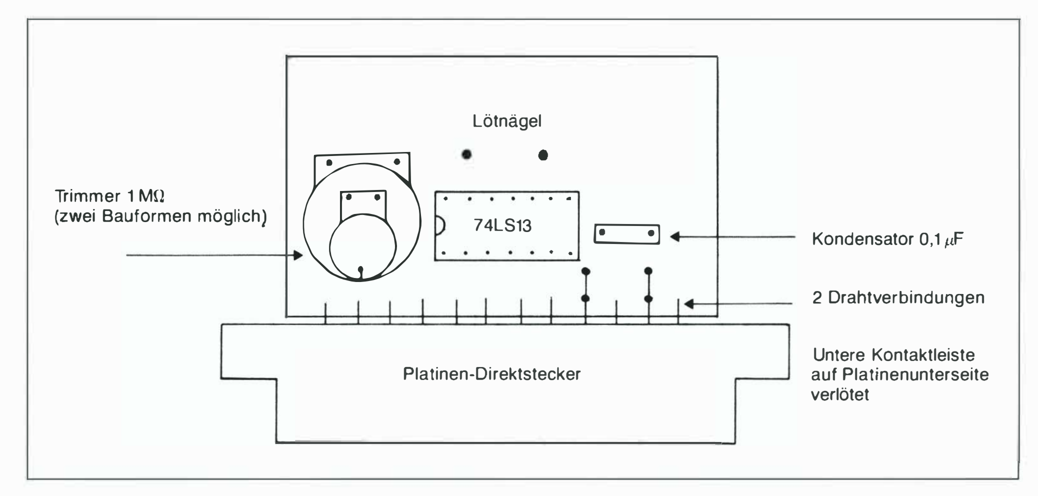


Bild 2.64: Bestückungsplan zur Frequenzmeter-Platine

in CIA # 2, die ja am NMI-Eingang des Prozessors angeschlossen ist, sowie durch den Befehl SEI (Set Interrupt Disable). Der Ausgang PB7 wird auf LOW gesetzt, um das UND-Gatter zu sperren, und die Fehlermeldung 0 für »Alles klar« wird in der Speicherzelle FLAG vorbereitet. Falls doch ein Fehler auftreten sollte (Überlauf oder Abbruch) wird sie dann wieder überschrieben. Die Bits im Interruptregister werden ebenfalls durch Lesen gelöscht.

Die einzige Interruptquelle ist jetzt noch die RESTORE-Taste, die ja ebenfalls auf den NMI-Eingang des Prozessors wirkt. Der NMI-Zeiger in den Speicherzellen $0318 und $0319 wird auf eine geänderte NMI-Routine verbogen, die weiter unten erläutert wird. Anschließend löscht das Programm den Zähler in CIA # 1, der später das Meßergebnis enthalten soll. Dies geschieht durch Überschreiben der zugehörigen Latchregister mit $FF und Setzen von Bit 4 in den Kontrollregistern. Außerdem wird durch die übrigen Bits gleich folgende Betriebsart eingestellt:

* Triggerquelle für A ist Anschluß CNT1 am User-Port
* Timer B zählt Unterläufe von Timer A, also Betrieb als voller 32-Bit-Zähler
* keine Unterlaufsignalisierung am Port
* sofortige Freigabe zum Zählen, sobald Impulse kommen

In ähnlicher Weise wird nun CIA # 2 initialisiert. Die Timer-Inhalte werden dabei durch Setzen von Bit 4 auf 0 festgesetzt, der Timer aber nicht gestartet. Nun erhalten die Latches Werte, die eine Unterlaufzeit von genau einer Sekunde ergeben. Es erfolgt keine Übernahme in den Zähler selbst, da Bit 4 nun 0 bleibt. Für die Timer in CIA # 2 wird folgende Betriebsart eingestellt:

* Triggerquelle für A ist der Systemtakt
* Timer B zählt Unterläufe von Timer A, also auch hier Betrieb als voller 32-Bit-Zähler
* Unterlaufsignalisierung von Timer B durch Kippen des Ausgangs PB7 (am User-Port) in die jeweils andere Lage
* sofortige Freigabe zum Zählen

Sobald das Kontrollregister für Timer A beschrieben wurde, startet der Zähler. Da sein Inhalt zunächst 0 ist, erzeugt der erste Taktimpuls nach dem Start bereits eine Zustandsänderung von Pin PB7, an dem ein Eingang des UND-Gatters angeschlossen ist, nämlich Kippen von LOW nach HIGH: Das »Tor« wird geöffnet. Nun wird das Zählerregister selbsttätig mit den in den Latches $DD04 bis $DD07 gespeicherten Werten geladen und benötigt genau eine Sekunde bis zum erneuten Unterlauf, bei dem PB7 wieder LOW wird. Während dieser Zeit bewirkt jeder Impuls am Meßeingang des UND-Gatters eine Änderung am User-Portpin CNT1 und erniedrigt den Zähler in CIA #1.

Da PB7 auch mit dem User-Port-Eingang $\overline{\text{FLAG2}}$ verbunden ist, setzt die fallende Flanke am Ende der Meßzeit das Flag-Interrupt-Bit im Register $DD0D. Dieser Vorgang wird durch die Schleife LOOP abgefragt, damit der Computer das Ende der Meßzeit erkennt. Er liest dann zuerst Bit 1 des Interrupt-Kontrollregisters von CIA # 1 aus. Dies wurde vor der Schleife durch Lesen gelöscht. Es kann also nur gesetzt sein, wenn ein Unterlauf des gesamten Zählregisters stattgefunden hat. In einem solchen Fall wäre das Meßergebnis falsch. Daher wird gegebenenfalls die Fehlermarke FLAG auf 2 gesetzt.

Weil der Zähler nach Rückgabe der Kontrolle an BASIC wieder andere Funktionen wahrnehmen muß, werden die Endwerte in den vier auf FLAG folgenden Speicherzellen festgehalten und an das übergeordnete Programm übergeben. Dann werden die CIAs mit einer Betriebssystemroutine wieder initialisiert, der normale Interruptbetrieb mit CLI freigegeben, und es erfolgt der Rücksprung. Sollte während des Programmlaufs die RESTORE-Taste betätigt werden, dann unterbricht der Rechner seine Tätigkeit und verzweigt zum Programmteil NMI. Wie üblich, werden hier zunächst die Register gerettet. Dann wird Timer B in CIA # 2 – die höherwertige Hälfte des Zählers, mit dem die Meßzeit erzeugt wird – auf 0 gesetzt und in FLAG der Fehlercode 1 (für »Abbruch«) geschrieben. Die Register werden wieder zurückgeholt und die normale Routine fortgeführt. Damit erfährt der Timer in CIA # 2 bereits nach sehr kurzer Zeit einen Unterlauf, und der Meßzyklus wird abgebrochen. Während einer Messung, die ja bei entsprechender Einstellung vom Hauptprogramm aus unter Umständen ziemlich lange dauern kann, stellt dies die einzige Möglichkeit dar, das Maschinenprogramm schnell zu verlassen.

2.4.4.5.2 Der BASIC-Teil

Das BASIC-Programm übernimmt im wesentlichen die Umrechnung des Zählergebnisses in die Anzeigewerte, sowie die optische Gestaltung des Bildschirms. Dazu wird der übergebene Zählerinhalt durch die Meßzeit geteilt. Es ergibt sich der Frequenzwert in Hertz. Die Periodendauer ist die für eine einzige Schwingung benötigte Zeit. Sie ist einfach der Kehrwert der Frequenz und wird in S für Sekunden, MS für Millisekunden, US für Mikrosekunden oder NS für Nanosekunden angegeben. Aus dem Zählerinhalt wird außerdem noch der maximale Systematische Fehler und die sich daraus ergebende Toleranz der beiden Anzeigewerte errechnet und angezeigt.

Als Besonderheit enthält die Software eine Routine zur Berechnung des Mittelwertes aus den letzten bis zu 99 Messungen. Dies ist vor allem bei langen Torzeiten und leicht jitternden Meßgrößen vorteilhaft. Nach dem Programmstart oder bei neu eingestellten Werten berechnet die Routine zunächst immer nur den Mittelwert aus den bis zu diesem Zeitpunkt gesammelten Daten so lange, bis genügend Meßergebnisse vorliegen.

2.4.4.6 Eichung

Leider bleibt Ihnen das nicht erspart. Wie bereits erwähnt, hängt die Meßgenauigkeit der Anordnung von der Torzeit und damit wieder vom Systemtakt des C 64 ab. Aus Kapitel 2.4.1.5 wissen Sie, daß diese Frequenz bei jedem C 64 leicht unterschiedlich ist. Um verläßliche Ergebnisse zu erhalten, müssen Sie also zunächst – wie dort beschrieben – die genaue Arbeitsfrequenz Ihres Computers bestimmen.

Laden Sie danach das Programm TORZ.TIMERWERTE und setzen Sie die gemessene Frequenz (in Hertz) dort in Zeile 30 ein. Nach dem Start mit RUN werden Sie gefragt, für welche Torzeit (in Sekunden) Sie die Werte berechnet haben wollen.

Antworten Sie zunächst mit 1 (RETURN). Das Programm berechnet die optimalen Timerwerte. Falls es keine genaue Lösung (ohne Abweichung) gibt, wählt es diejenige mit dem geringsten Rest. In diesem Fall müssen aber alle 65 536 Kombinationsmöglichkeiten der beiden Timer A und B durchlaufen werden. Das kann eine Weile dauern. Zu Ihrer Information werden auf dem Bildschirm jeweils die aktuellen Werte und der bisher berechnete kleinste Rest ausgegeben. Falls Ihnen die Genauigkeit genügt, können Sie das Programm jederzeit abbrechen und mit GOTO 2000 die Anzeige der berechneten Werte aufrufen.

Die angezeigten Daten für A Low, A High, B Low und B High sind dann in das Meßprogramm zu übertragen. Für eine Sekunde ist die Zeile 3070 zuständig. Den Wert für A Low schreiben Sie nach AL(4)= , den für A High nach AH(4)= und so fort.

Wiederholen Sie den gleichen Vorgang auch für die übrigen Torzeiten.

Nun ist Ihr Frequenzzähler geeicht und gebrauchsfertig.

2.4.4.7 Bedienung des Frequenzmeßgerätes

Der Meßzusatz darf nur bei ausgeschaltetem Computer auf den User-Port-Anschluß aufgesteckt werden. Nach dem Einschalten wird das BASIC-Programm normal geladen und mit RUN gestartet. Es meldet sich mit dem Anzeigebild. Die Torzeit ist zunächst auf eine Sekunde eingestellt, und der Mittelwert wird aus den letzten fünf Messungen berechnet. Sollen diese Werte geändert werden, ist die RESTORE-Taste zu betätigen. Der untere Bildschirmteil wird aufgehellt und die Bedienungstasten angezeigt. So kann die Torzeit mit Hilfe der Tasten »+« und »-« verändert werden, sowie die Anzahl der Messungen für den Mittelwert mit dem »kleiner« und »größer« Zeichen, egal ob mit SHIFT oder ohne. Dabei springt der Wert auch von 1 direkt nach 99 und umgekehrt. Größte Genauigkeit erhalten Sie bei langen Torzeiten.

Bei TTL-Eingangspegeln sollte das Trimmpoti ganz aufgedreht sein. Wie beschrieben, lassen sich mit der Impulsformerschaltung nach Bild 2.63 auch beliebige andere Spannungen verarbeiten. Der Experimentierfreude sind keine Grenzen gesetzt, und sollte wirklich etwas zerstört werden, dann wird es nicht etwa der Computer sein, sondern schlimmstenfalls das IC mit dem UND-Gatter, das billig und leicht auswechselbar ist.

2.4.4.8 Frequenzmeter als Drehzahlmesser

Frequenzen tauchen in der Technik und in der Natur überall auf. Eine Sonderform der Frequenz ist die Drehzahl. Wollen Sie beispielsweise die Drehzahl eines Motors messen, dann braucht man einen geeigneten Impulsformer. Beim Motor eines Modellflugzeugs mit aufgesetzter Luftschraube ist das kein Problem. Sie können sich zum Beispiel unseres Lichtschrankenmoduls aus Kapitel 2.2.4.4 bedienen. Immer, wenn ein Blatt der Luftschraube die Lichtschranke unterbricht, sendet der Modul einen Impuls aus. Es entsteht eine Frequenz, die ein Vielfaches der Umdrehungszahl ist.
Eine zweiblättrige Luftschraube erzeugt beispielsweise zwei Signale pro Umdrehung. Die angezeigte Frequenz ist somit durch zwei zu teilen, um auf die Drehzahl zu kommen.
Auch hier gilt: Je mehr Impulse pro Umdrehung erzeugt werden, um so genauer kann das Ergebnis ermittelt werden. Will man die relativ langsame Drehzahl eines Plattentellers messen, ist es ratsam, als Impulsgeber eine Scheibe mit vielen gleichmäßig auf dem Umfang angebrachten Markierungen aufzulegen und mit einer Reflexlichtschranke abzutasten.

2.4.5 Geiger-Müller-Zähler

2.4.5.1 Radioaktivität messen

Im Fall des Reaktorunglücks von Tschernobyl stifteten vor allem voneinander abweichende Angaben über radioaktive Belastung von Umwelt und Lebensmitteln große Verwirrung. Glücklich war derjenige, der zu Hause selbst nachmessen konnte. Es ist in jedem Fall gut, ein Meßgerät zur Hand zu haben, denn radioaktive Teilchen werden unterschiedlich dicht in die Wachsschicht von Pflanzenblättern, Obst und Gemüse eingebaut.
Der hier vorgestellte Geiger-Müller-Zähler kann Sie warnen. Was Sie jedoch daraufhin unternehmen, das müssen Sie selbst wissen...
Für das menschliche Auge sind radioaktive Strahlen unsichtbar. Um ihre Eigenschaften erforschen zu können, mußten zunächst spezielle Geräte entwickelt werden. Wegen ihrer hohen Energie sind alle drei Strahlungsarten (Alpha-, Beta- und Gammastrahlen) in der Lage, Gase zu ionisieren. Diese Eigenschaft wird in einem Strahlungsmeßgerät ausgenutzt, das nach seinen Erfindern Geiger-Müllersches-Zählrohr genannt wird. Sein Aufbau ist im Prinzip sehr einfach. Es besteht aus einem dünnwandigen, mit Edelgas gefüllten Metallrohr, in dessen Mittelachse sich ein elektrisch isoliert gespannter Draht befindet. Zwischen Gehäuse und Draht wird eine hohe Gleichspannung angelegt.
Tritt nun ein Strahlungsteilchen in das In-

nere des Zählrohrs ein, dann werden Elektronen frei, die sich aufgrund des hohen elektrischen Feldes stark beschleunigt auf den Mitteldraht zubewegen und dabei eine ganze Lawine anderer elektrisch geladener Teilchen mitreißen. Dadurch entsteht im Draht selbst ein kurzer Stromstoß, der über einen Widerstand in einen Spannungsimpuls umgewandelt wird, und mit einer empfindlichen Elektronik ausgewertet werden kann.

Bild 2.65 zeigt eine entsprechende Schaltung zum Anschluß an den C64.

2.4.5.2 Die Meßschaltung

Sehen wir uns zunächst an, auf welche Weise die Hochspannung für die Röhre erzeugt wird. Die Schaltung ist zum Anschluß an den User-Port gedacht. Glücklicherweise existiert dort eine Wechselspannung von 9 Volt, mit der ganz unterschiedliche Hilfsspannungen erzeugt werden können. In diesem Fall wurde ein 6-V-Netztrafo eingesetzt, der üblicherweise die 220-V-Spannung aus der Steckdose heruntertransformiert. Dieser Effekt kann auch umgekehrt benutzt werden, indem die 6-V-Wicklung an die 9-V-Wechselspannung gelegt wird. Am Ausgang des

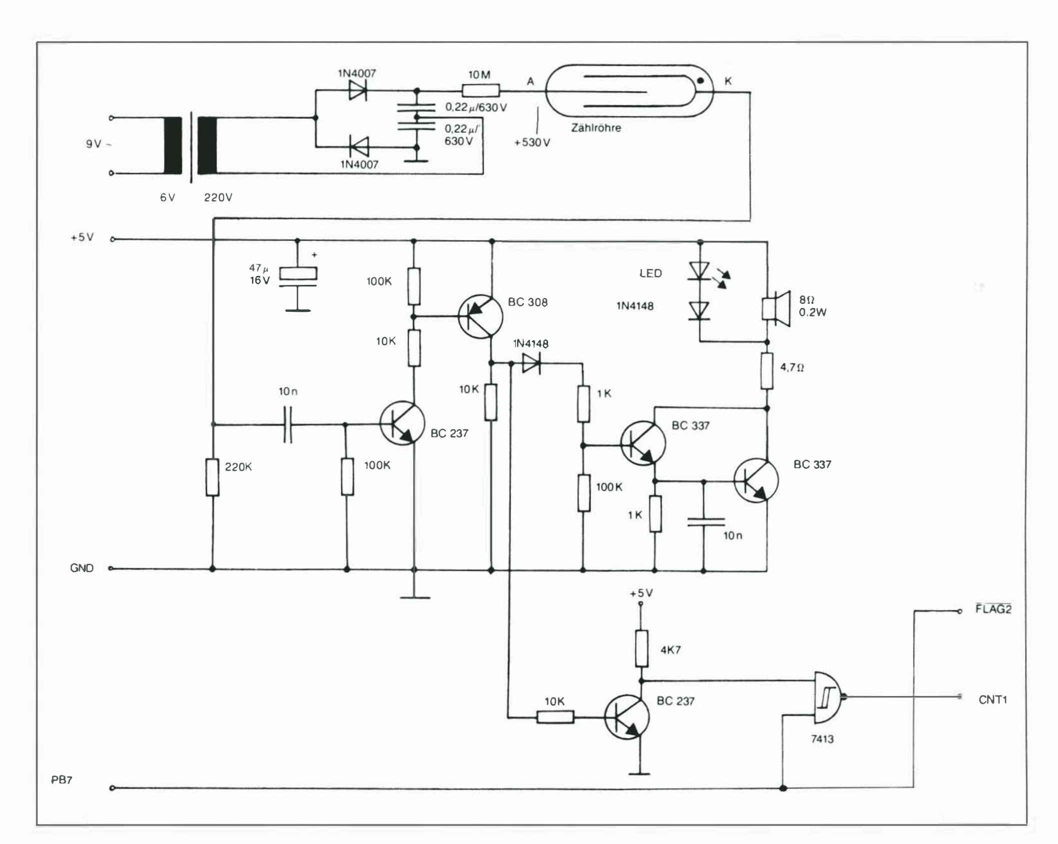


Bild 2.65: Schaltplan des Geiger-Müller-Zählers

Trafos entsteht so eine Spannung von etwa 200 V_{eff}, deren positive und negative Halbwelle zwei Kondensatoren jeweils über eine Diode auf den Spitzenwert aufladen. Diese Kondensatoren werden in Reihe geschaltet und die abgegriffene Spannung von etwa 530 Volt über einen Schutzwiderstand an die Röhre gelegt. Da praktisch kein Strom durch die Röhre fließt, fällt an dem sehr hochohmigen Widerstand auch keine Spannung ab.
Die in der Röhre entstehenden Zählimpulse haben durch den Lawineneffekt immer eine Länge von etwa 0,1ms. Sie werden über einen Kondensator in die Auswerteschaltung eingekoppelt. Am Kollektor des zweiten Transistors verzweigt die Information. Der obere Schaltungsteil leitet die Impulse über einen kleinen Leistungsteil auf eine Leuchtdiode zur optischen und auf einen Lautsprecher zur akustischen Kontrolle. Dabei kann der Lautsprecher auch abschaltbar gemacht werden.
Der Transistor im unteren Zweig schaltet die Impulse auf ein UND-Gatter weiter, das Ihnen bereits vom Frequenzzähler her bekannt vorkommen dürfte. Das ist auch nicht weiter verwunderlich, denn schließlich steht man hier vor dem gleichen Problem, nämlich die Anzahl von Impulsen während einer bestimmten Zeit zu ermitteln. Diese Impulsrate läßt Rückschlüsse auf die Strahlenbelastung zu.

2.4.5.3 Geigerzähler im Eigenbau

Bild 2.66 zeigt die Platine der Meßschaltung nach Bild 2.65. Der zugehörige Bestückungsplan ist in Bild 2.67 wiedergegeben und Tabelle 2.26 enthält die benötigten Bauteile. Beim Bestücken richten Sie sich bitte wieder nach den Hinweisen im Anhang A.

1	Zählrohr (z.B. ZP 1200 von Valvo)
1	Logik-IC 74LS13
1	Sockel 14polig
2	Transistoren BC237
1	Transistor BC308
2	Transistoren BC337 bzw. BC338
2	Dioden 1N4007
2	Dioden 1N4148
1	Leuchtdiode, rot
1	Trafo 220 Volt / 6 Volt
1	Widerstand 10 Megaohm
1	Widerstand 220 Kiloohm
3	Widerstände 100 Kiloohm
3	Widerstände 10 Kiloohm
1	Widerstand 4,7 Kiloohm
2	Widerstände 1 Kiloohm
1	Widerstand 4,7 Ohm
2	Kondensatoren 0,22 Mikrofarad / 630 Volt
1	Elektrolytkondensator 47 Mikrofarad / 16 Volt, radial
2	Kondensatoren 10 Nanofarad
1	Miniaturlautsprecher 8 Ohm
1	einseitige Platine nach Bild 2.66
4	Lötnägel
1	Bügel für Zugentlastung mit Befestigungsmaterial
1	passendes Gehäuse
1	Kabel 6polig
1	User-Port-Stecker mit Gehäuse

Tabelle 2.26: Bauteile zum Geigerzähler

Wegen der hohen Spannung für die Röhre muß die Platine unbedingt in ein Gehäuse eingebaut werden. Im Handel gibt es eine Vielzahl geeigneter Ausführungen.

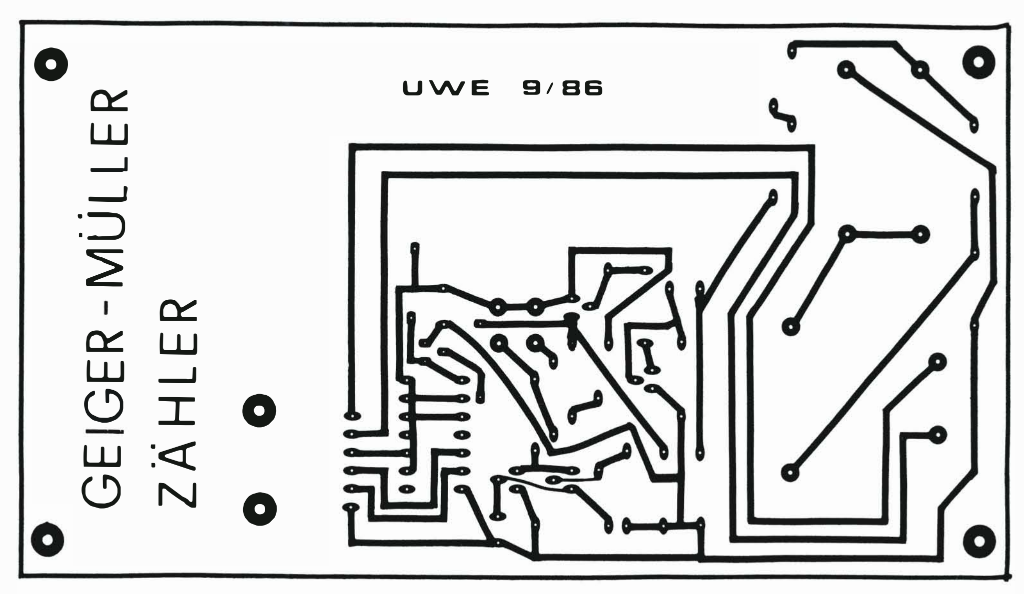


Bild 2.66: Platinenvorlage für einen Geigerzähler

Bild 2.67: Bestückungsplan des Geigerzählers

Der Lautsprecher und die Leuchtdiode können im Deckel des Gehäuses befestigt werden. Im Mustergerät wurden zunächst Löcher gebohrt und die Elemente dann eingeklebt.
Der Anschluß des C64 erfolgt über ein Kabel am User-Port. Auf der Platine ist bereits eine Zugentlastung mittels eines Metallbügels vorgesehen.
Zum Schutz gegen Beschädigung sollte die Röhre in Schaumstoff gehüllt und beispielsweise mit Klebeband ebenfalls an der Platine befestigt werden. Foto 2.10 zeigt die fertige Anordnung im Gehäuse.

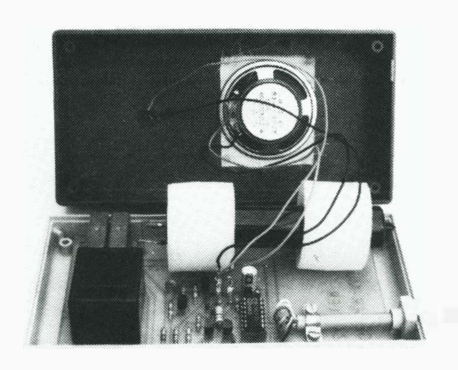


Foto 2.10: Geigerzähler für den C 64

2.4.5.4 Ermittlung der Strahlenbelastung

Nach jeder Entladung bleibt das Zählrohr für eine gewisse Zeit lang unempfindlich. Infolge dieser Totzeit ist jeder gemessene Impuls auch nur ein Maß für viele, tatsächlich in der Luft befindliche Teilchen. Dazu kommt noch die Empfindlichkeit der Röhre. Im Arbeitsbereich des Zählrohrs verhält sich dieser Zusammenhang linear, das heißt, es gibt einen Faktor, mit dem die gezählten Impulse zu multiplizieren sind, um auf die Dosisrate in Rem pro Stunde zu kommen. Dieser Faktor ist bei jedem Zählrohr und bei jeder Einzelanwendung verschieden. Bei der Dosisrate spielen auch noch die Energien der einzelnen gemessenen Isotope eine Rolle, so daß tatsächlich die Angabe von Impulsen pro Sekunde wissenschaftlich am sinnvollsten sein dürfte.
Die Empfindlichkeit der meisten Geiger-Müller-Zählrohre liegt aber etwa bei 20%, so daß der Faktor 5 angenommen werden kann, um einen ersten Eindruck zu bekommen. Der Fachhandel – oder im Zweifelsfall der Hersteller der Röhre – kann sicherlich genauere Angaben für spezielle Einzelfälle machen.
Das Meßprogramm GEIGERZAEHLER kann seine Verwandtschaft zum Frequenzzähler nicht leugnen. Es ist auch tatsächlich aus diesem Programm durch einige Änderungen entstanden. Zu Anfang kann über die Variable PF der eben erwähnte Proportionalitätsfaktor der benutzten Röhre abgeändert werden.
Die Anzeige erfolgt in Impulsen pro Sekunde (/S) und in Millirem pro Stunde (MR/H). Außerdem wird wieder ein Mittelwert gebildet, der wegen der statistischen Ereignisverteilung allein verläßlichen Aussagewert haben kann.
Das Ändern der Einstellungen geschieht auch hier nach Drücken von RESTORE. Daraufhin wird der untere Bildschirmteil aufgehellt und man kann nach wie vor die Torzeit mit den Tasten »+« und »-« ändern. Alle anderen Funktionen erreicht man jedoch durch Drücken der inversen Anfangsbuchstaben. Auch eine einstellbare Warnschwelle ist verfügbar. Sobald der angegebene Wert einmal kurz überschritten wurde, leuchtet der Rahmen des Bildschirms rot auf. Die

Warnschwelle bezieht sich immer auf den Mittelwert.
Wollen Sie den Rahmen wieder auf seine Standardfarbe zurückstellen, drücken Sie hintereinander RESTORE und RETURN. Mit RETURN kommen Sie immer wieder zum Anzeigemodus zurück.

2.4.5.5 Jetzt wird gemessen

Wie Sie sehen, zeigt die Anordnung auch ohne ein in der Nähe befindliches radioaktives Präparat bereits eine gewisse Zahl von Impulsen an. Es handelt sich dabei um den sogenannten Nulleffekt, der zum einen durch kosmische Strahlung und durch die immer vorhandene natürliche Umgebungsradioaktivität sowie zum anderen durch Störimpulse im Zähler selbst entsteht. Will man Strahlungsmessungen an bestimmten Objekten vornehmen, dann muß der Nulleffekt vorher experimentell ermittelt und vom Meßergebnis abgezogen werden.
Wahrscheinlich werden Sie auf Anhieb gar kein Objekt finden, um den Zähler ausprobieren zu können – und das ist ja eigentlich auch ganz beruhigend. Wollen Sie es trotzdem einmal schneller ticken hören, als durch die Hintergrundstrahlung hervorgerufen, dann besorgen Sie sich in einem Campingladen einfach einen Glühstrumpf. Erstaunlicherweise handelt es sich dabei um einen sehr schwach radioaktiven Alpha-, Beta- und Gammastrahler mit natürlichen radioaktiven Stoffen (Th_{232} und Ce_{142}). Es sind jedoch keine speziellen Vorsichtsmaßnahmen erforderlich.
Dieser Geiger-Müller-Zähler kann keine Labor- oder Industriemessung ersetzen. Er dient lediglich zum Nachweis von radioaktiven Strahlen bzw. von erhöhter Radioaktivität.

2.5 Der serielle Datenport

Sehr stiefmütterlich behandelt wird beim C 64 oft der serielle Port. Obwohl die entsprechenden Anschlüsse beider CIAs am User-Port herausgeführt sind, bleiben sie selbst bei der im Betriebssystem eingebundenen seriellen RS232-Schnittstelle unbenutzt.

2.5.1 Bit für Bit im Gänsemarsch

Der serielle Port taucht in der Programmierübersicht der CIAs (Tabelle im Anhang B) an drei Stellen auf. Zunächst bildet die Adresse $XX0C das serielle Datenregister. Bit 6 im Kontrollregister A ($XX0E) legt die Richtung der Datenübertragung fest und schließlich gibt Bit 3 des Interrupt-Kontrollregisters ($XX0D) den Zustand des Datenport wieder. Außerdem spielt noch Timer A eine Rolle.
Entgegen Angaben in anderen Büchern geschieht die Programmierung des seriellen Port auf Ausgang – wie bei allen anderen Portleitungen auch – durch Setzen des Richtungsbit auf 1. Nach dem Einschalten oder einem Reset ist dieses Bit immer gelöscht und der Port somit als Eingang definiert.
Der Pin SP am User-Port stellt den seriellen Anschluß dar, der Pin CNT führt das zugehörige Taktsignal.
Das serielle Datenregister ist ein 8 Bit breiter Pufferspeicher. Der eigentliche Datenaustausch geschieht über ein Schieberegister, das vom Programmierer her unzugänglich ist.

Falls der serielle Port als Eingang programmiert ist, fungiert auch CNT als Takteingang. Bei jeder positiven Flanke an CNT wird der Zustand von SP in das Schieberegister übernommen. Dabei wird der bisherige Inhalt dieses Registers nach links geschoben und das neue Bit rechts angefügt. Ist dies achtmal geschehen, dann wird der Wert des Schieberegisters in das serielle Datenregister übertragen. Gleichzeitig wird Bit 3 im Interrupt-Kontrollregister gesetzt, um anzuzeigen, daß ein neues Datenwort bereitsteht.
Im Ausgabemodus dient der Anschluß CNT als Taktausgang. Bei jedem Unterlauf von Timer A ändert er seinen Zustand. Mit Timer A wird also die Geschwindigkeit der Datenausgabe bestimmt. Da der kleinste Inhalt des Timer $0001 ist ($0000 erzeugt keinen Unterlauf!), beträgt die höchste Übertragungsrate ein Viertel des Systemtaktes. Schreibt man einen Wert in das serielle Datenregister, dann wird er in das Schieberegister übernommen, sobald dieses leer ist. Der Inhalt des Schieberegisters wird Bit für Bit auf den Anschluß SP gelegt und dann eine positive Flanke an CNT ausgegeben. Dabei ist auch hier die Schieberichtung wieder links, so daß das höchstwertige Bit zuerst erscheint. Sind alle 8 Bit herausgeschoben, so wird – analog zum Empfang von Daten – das Bit 3 im Interrupt-Kontrollregister gesetzt. Dies geschieht nicht, wenn bereits vor der Übertragung des achten Bit neue Daten ins serielle Datenregister geschrieben wurden. In diesem Fall würde die Übertragung direkt fortgesetzt. Falls kein neues Byte ins Datenregister geschrieben wird, hält die Datenausgabe an.

2.5.2 Erzeugung einer Gleichspannung aus Rechteckimpulsen

Erinnern Sie sich noch an das Problem, auf der SC-01-Sprachausgabekarte eine variable Gleichspannung zu erzeugen? Sie wissen jetzt, wie Sie Daten seriell ausgeben, doch wie erhält man damit eine Gleichspannung? Sehr einfach, denn man kann doch eine Rechteckspannung mit variabler Impulsbreite erzeugen. Hat diese Rechteckspannung eine genügend hohe Frequenz, dann zeigt ein Vielfachmeßgerät bereits einen konstanten Wert an. Das kommt daher, weil der Zeiger aufgrund seiner Trägheit den schnellen Spannungswechseln nicht mehr folgen kann: Er zeigt einfach einen Mittelwert an, bzw. er integriert den Spannungsverlauf mechanisch.
Analog dazu verhält sich elektrisch ein Tiefpaß, also eine Hintereinanderschaltung aus Widerstand und Kondensator nach Bild 2.68. Die Dimensionierung der Bauteile hängt von der Frequenz und der Belastung am Ausgang ab. Über den Widerstand wird ein Kondensator während der Spannungsimpulse aufgeladen. Er entlädt sich aber nur langsam, so daß sich ein konstanter Mittelwert einstellt.
Im C64 kann man die geforderte Frequenz zusätzlich zu der bereits beschriebenen Sprachausgabeschaltung am bequemsten generieren, indem während der Interrupt-Routine einfach das serielle Datenregister des CIA # 1 mit einem speziellen Bitmuster geladen wird. Timer A wird zur Erzeugung des System-Interrupt benötigt und läuft bereits. Der nötige Zusatz in der IRQ-Routine lautet lediglich:

```
LDA $02     ; Speicherzelle 2 einlesen
STA $DC0C   ; ins serielle Datenregi-
              ster bringen
JMP $EA31   ; und weiter mit norma-
              ler IRQ-Routine
```

Die normalerweise freie Adresse 2 wird dabei zur Übergabe der Bitkombinationen für die Spannungserzeugung verwendet. Folgende Eingabe ändert bereits die Tonhöhe:

```
POKE 2,X ,
```

wobei X eine der Kombinationen aus Tabelle 2.27 darstellt.

Kombination	X
00000000	0
00000001	1
00000011	3
00000111	7
00001111	15
00011111	31
00111111	63
01111111	127
11111111	255

Tabelle 2.27: Die Bitkombinationen zur Spannungserzeugung

Leider ist man bei dieser Methode auf eine relativ niedrige Frequenz angewiesen. Das heißt, um eine genügend glatte Spannung zu erhalten, muß der Kondensator recht groß gewählt und eine relativ lange Wandlungszeit in Kauf genommen werden. Im Schaltplan der Sprachausgabekarte (Bild 2.51) liegt der Tiefpaß ganz oben. Zusätzlich wurde eine Diode eingefügt, die verhindern soll, daß sich der Kondensator bei LOW über den seriellen Port entlädt.

2.6 RS232

Sehr viele kommerzielle und auch immer mehr Heimgeräte verfügen heutzutage

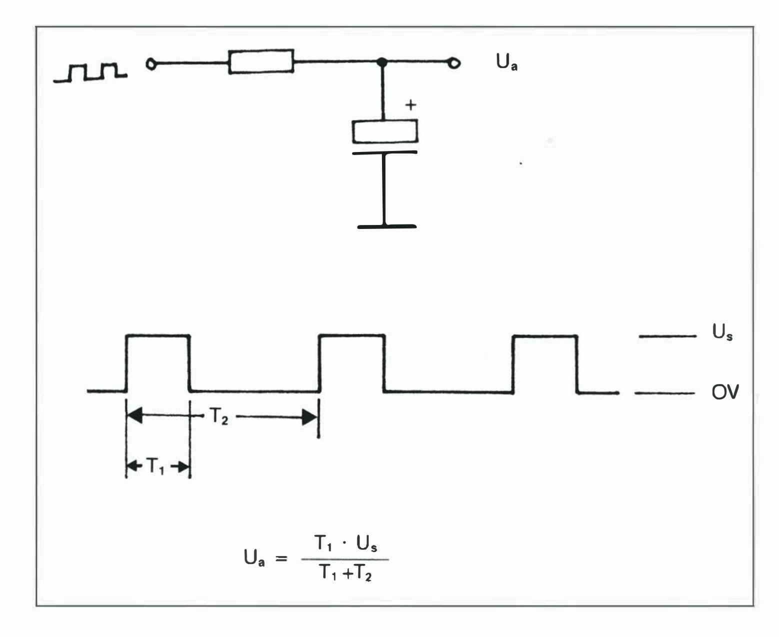


Bild 2.68: Umsetzung einer Rechteckspannung und eine Gleichspannung mit einem Tiefpaß

über eine RS232-Schnittstelle zur Datenübertragung. Was damit anzufangen ist, legt der folgende Abschnitt dar.

2.6.1 Eine Allerweltsschnittstelle

2.6.1.1 Grundideen

Es war schon die Rede davon, daß der User-Port des C64 auch als RS232-Schnittstelle eingesetzt werden kann. Im Gegensatz zur parallelen Centronics-Norm ist die Grundidee bei dieser seriellen Verbindungsart, die Daten nicht byteweise, sondern Bit für Bit als Spannungsimpulse zu übermitteln. Dies geschieht in einer vorher genau festgelegten Geschwindigkeit, damit der Empfänger den Sender überhaupt verstehen kann. Die Übermittlungsgeschwindigkeit wird in Baud gemessen. Ein Baud ist genau ein Bit in einer Sekunde. Tabelle 2.28 gibt die gebräuchlichen Baudraten an.

110	Bit/s
150	Bit/s
300	Bit/s
600	Bit/s
1200	Bit/s
2400	Bit/s
4800	Bit/s
9600	Bit/s
19200	Bit/s

Tabelle 2.28: Übliche Baudraten bei serieller Übertragung

Um Sender und Empfänger zeitlich exakt aufeinander abstimmen (synchronisieren) zu können, werden in der Praxis Pakete von 5 bis 8 Datenbit übertragen, die von einem Startbit und 1 bis 2 Stoppbit eingerahmt sind. Das Startbit hat grundsätzlich LOW- und das Stoppbit HIGH-Pegel. Dadurch entsteht bei der Übertragung zwischen 2 Byte immer eine fallende Flanke, aus der eine geeignete Vorrichtung im Empfänger (Soft- oder Hardware) den Sendetakt zurückgewinnen kann. Eine Datenübertragung mit einer einzigen zweiadrigen Leitung ist damit durchaus möglich.

Vor dem Stoppbit kann zur Fehlererkennung noch ein sogenanntes Paritäts-Bit vereinbart werden, das die Anzahl der HIGH-Zustände im Datenpaket immer gerade oder ungerade macht. Sind zum Beispiel in einer 8-Bit-Übertragung 5 Bit HIGH, wird das Paritäts-Bit vom Sender ebenfalls gesetzt, falls gerade Parität vereinbart wurde. Im Empfänger wird der Zustand der Bits überprüft und Einzelfehler können erkannt und angezeigt werden.

Um die Störanfälligkeit besonders bei Langstreckenübertragungen zu vermindern, wurde bei der RS232-Schnittstelle von den sonst üblichen TTL-Pegeln abgewichen. Schon der Leitungswiderstand des Kabels würde auf langen Strecken nämlich bewirken, daß ausgesendete HIGH-Pegel beim Empfänger nicht mehr mit 5 Volt ankämen, sondern vielleicht nur noch mit 2 Volt, was bereits als ein LOW-Signal interpretiert werden müßte.

Bei der RS232-Norm wurde sich aus diesem Grund auf Spannungen von –12 bis –3 Volt für ein gesetztes Bit und +3 bis +12 Volt für eine logische Null geeinigt. Wegen der sich um maximal 24 Volt unterscheidenden Potentiale wurde die RS232- Übertragung hierzulande V.24-Norm getauft (DIN 66020). Allein die Definition, welche Spannung für 1 und welche für 0 steht, ist unterschiedlich, was

allgemein zu einiger Verwirrung führt. Hier kann nur der Rat gegeben werden: Nicht auf Beipackinformationen verlassen, sondern im Einzelfall selbst nachprüfen! Auch bei den Herstellern wird oft manches durcheinandergeworfen. Im Falle eines Falles helfen zwischengeschaltete Inverter weiter.

2.6.1.2 Von Leitungen und Übertragungsprotokollen

Wie erläutert, ist die Übertragung von Daten grundsätzlich mit den dargestellten Vereinbarungen über eine nur zweiadrige Verbindung möglich. Die Bezeichnung der Sendeleitung ist allgemein TXD (Transmit Data = Sendedaten); die andere Ader stellt natürlich die Masseverbindung her.
Sollen gleichzeitig auch Daten zurückgeschickt werden, wird eine dritte Ader nötig. Ihre Bezeichnung ist gewöhnlich RXD (Receive Data = Empfangsdaten). Damit können auch Geräte bedient werden, die nicht so schnell arbeiten wie der Sender, indem auf der Rückleitung ein bestimmtes Zeichens übermittelt wird, um den Sender erst einmal zu stoppen. Es handelt sich um ein sogenanntes Softwareprotokoll, wovon das gebräuchlichste das XON-XOFF-Protokoll ist. Dabei werden die beiden ASCII-Steuerzeichen DC3 (XOFF = $13 = CTRL-S) und DC1 (XON = $11 = CTRL-Q) verwendet. Mit XOFF stoppt der Datenempfänger die Sendung seiner Gegenstation, während er intern zum Beispiel mit Ausdrucken oder Abspeichern beschäftigt ist, mit XON dagegen meldet er, daß er wieder zur Aufnahme weiterer Zeichen bereit ist.
Sender und Empfänger müssen dann allerdings in der Lage sein, gleichzeitig zu senden und zu empfangen.
Es geht auch anders, wie die EXT-ACK-Prozedur zeigt. Der Computer sendet hier einen ganzen Datenblock, der mit EXT (End Of Text = Textende = $03) abgeschlossen wird. Sobald er von der Gegenstation komplett verarbeitet ist, sendet diese das Steuerzeichen ACK (Acknowledge = Bestätigung = $06) zurück, woran der Computer erkennt, daß er jetzt den nächsten Datenblock senden kann.
Die EXT-ACK-Prozedur setzt mindestens empfangsseitig einen Pufferspeicher voraus, da der Empfänger ja die Übertragung während eines Datenblocks nicht anhalten kann.
Falls es die Hardware erlaubt, ist es meist doch wieder einfacher, eine zusätzliche Handshake-Leitung zu realisieren: DTR (Data Terminal Ready = Datenempfänger bereit). Kann der Empfänger keine Daten mehr aufnehmen, braucht er lediglich diese Leitung zu aktivieren. Der Sender wartet daraufhin so lange, bis das Signal wieder zurückgenommen wird.
Besonders bei Übertragungen mit hohen Baudraten über sehr lange Kabel oder in gestörten Umgebungen kann es schwierig sein, den Takt aus den seriellen Daten zurückzugewinnen. Manche Geräte besitzen auch gar keine eigenen Taktgeber. Es ist dann sinnvoll, die Synchronisationsimpulse auf einer separaten Ader einzeln mit zu übertragen. Dafür sind die Anschlüsse TC (Transmit Clock = Sendetakt) und RC (Receiver Signal Clock) miteinander zu verbinden.

Pin	Signal	Name	EIA	Abk.	Richtg.	Betr.art
A	GND	Protective Ground	AA	GND	-	1 2
B	FLAG2	Received Data	BB	Sin	ein	1 2
C	PB0	Received Data	BB	Sin	ein	1 2
D	PB1	Request To Send	CA	RTS	aus	1 2
E	PB2	Data Terminal Ready	CD	DTR	aus	1 2
F	PB3	Ring Indicator	CE	RI	ein	3
H	PB4	Received Line Signal	CF	DCD	ein	2
J	PB5				ein	3
K	PB6	Clear To Send	CB	CTS	ein	2
L	PB7	Data Set Ready	CC	DSR	ein	2
M	PA2	Transmitted Data	BA	Sout	aus	1 2
A	GND	Signal Ground	AA	GND	-	1 2 3

Tabelle 2.29: Pinbezeichnung der User-Portpins bei RS232-Betrieb

2.6.1.3 Die RS232-Schnittstelle des C64

Die eben aufgeführten Leitungen kann auch der C64 mit seiner eingebauten Schnittstellensoftware handhaben. Dazu werden diverse Anschlüsse am User-Port benutzt. Tabelle 2.29 zeigt, welche Funktionen die einzelnen Pins bei RS232-Betrieb haben.

Es wurde bereits erwähnt, daß der C64 nicht die seriellen CIA-Ports für seine RS232-Handhabung benutzt. Vielmehr installiert das Betriebssystem über CIA # 2 einen NMI-Betrieb höchster Ordnung, der verdeckt hinter BASIC- oder Maschinenprogrammen abläuft. Um den Verlust von Daten zu verhindern, werden zu Anfang der Übertragung zwei FIFO-Pufferspeicher generiert, die jeweils 256 Byte umfassen. Trotzdem können während des Kassettenbetriebs oder bei Aktionen auf dem seriellen Bus keine RS232-Daten empfangen werden. Bei umsichtiger Programmierung sollte das aber keine Probleme bereiten.

Der C64 verwaltet auch die RS232-Schnittstelle als Peripherieeinheit. Sie kann als Gerät mit der Nummer 2 angesprochen werden. Zur Bestimmung der Kontrollparameter sind zwei Register vorhanden, die auch von BASIC aus zugänglich sind. Tabelle 2.30 zeigt, welche Funktion die einzelnen Bits in diesen Registern erfüllen.

Mit Hilfe dieser Tabelle können Sie die Werte bestimmen, die Sie in die beiden Register schreiben müssen, um ein bestimmtes Übertragungsprotokoll zu erreichen. Das Aktivieren der RS232-Schnittstelle geschieht dann mit

```
OPEN Filenummer,
2,0,CHR$(Kontrollregister)
+CHR$(Kommandoregister)
```

Kontrollregister:

Bit	Inhalt	Wert	Funktion
3,2,1,0	0000	0	selbstdefinierte Übertragung
	0001	1	50 Bit/s
	0010	2	75 Bit/s
	0011	3	110 Bit/s
	0100	4	134,5 Bit/s
	0101	5	150 Bit/s
	0110	6	300 Bit/s
	0111	7	600 Bit/s
	1000	8	1200 Bit/s
	1001	9	1800 Bit/s
	1010	10	2400 Bit/s
4	X	0	(nicht benutzt)
6,5	00	0	8 Datenbit
	01	32	7 Datenbit
	10	64	6 Datenbit
	11	96	5 Datenbit
7	0	0	1 Stoppbit
	1	128	2 Stoppbit
Kommandoregister:			
Bit	Inhalt	Wert	Funktion
0	0	0	3-Line (Software-Protokoll)
	1	1	X-Line (Hardware-Protokoll)
3,2,1	XXX	0	(nicht benutzt)
4	0	0	Vollduplex
	1	16	Halbduplex
7,6,5	000	0	keine Paritätsüberprüfung
	001	32	ungerade Parität
	010	64	keine Paritätsüberprüfung
	011	96	gerade Parität
	100	128	keine Paritätsüberprüfung
	101	160	8. Datenbit = 1, keine Parität
	110	192	keine Paritätsüberprüfung
	111	224	8. Datenbit = 0, keine Parität

Tabelle 2.30: Die Register der RS232-Schnittstelle

Bit	Wertigkeit	Bedeutung
0	1	Paritätsfehler
1	2	Rahmenfehler (Start-/Stoppbit)
2	4	Empfängerpuffer voll
3	8	(unbenutzt)
4	16	CTS-Signal fehlt
5	32	(unbenutzt)
6	64	DSR-Signal fehlt
7	128	Break-Signal empfangen

Tabelle 2.31: Bedeutung der Statusbits

Die in der Tabelle angegebenen Dezimalwerte der gewünschten Merkmale werden einfach addiert. Beispielsweise legt man durch

```
OPEN 1,2,0,CHR$(6+64+0)
+CHR$(1+0+32)
```

für Filenummer 1 eine Vollduplexübertragung mit 1 Stoppbit, 6 Datenbit, 300 Baud, ungerader Parität und Hardwareprotokoll fest.

Wird eine Filenummer größer als 128 verwendet, sendet der Computer nach jedem Carriage Return (= Wagenrücklauf $0D) ein Linefeed (= Zeilenvorschub $0A).

Das Betriebssystem des C64 verfügt auch über eine Fehlerbehandlung der RS232-Schnittstelle, die entweder von BASIC aus mittels der Statusvariablen ST oder direkt durch Auslesen der Speicherzelle $0297 (= dezimal 663) abgefragt werden kann. Tabelle 2.31 enthält die Bedeutung der einzelnen Bits.

2.6.2 Pegel berichtigt

Leider gibt der C64 am User-Port auch bei RS232-Betrieb nur TTL-Pegel aus, so daß zumindest in diesem Punkt die Norm nicht voll erfüllt wird, doch gibt es inzwischen integrierte Bausteine, die alle Signale ohne Probleme normgerecht umformen. Üblicherweise benötigen sie dazu eine symmetrische Versorgungsspannung mit +12 Volt und –12 Volt. Eine Ausnahme bildet der MAX 232, der sich mit einer einfachen Versorgungsspannung von +5 Volt begnügt.

Bild 2.69 zeigt ein Blockdiagramm zu seiner Funktion. Im oberen Teil wurde ein Spannungswandler angedeutet, der die benötigten Hilfsspannungen über vier externe Elkos aus +5 Volt erzeugt. Im unteren Teil sind vier invertierende Treiber zu erkennen, von denen zwei TTL-Ein- und RS232-Ausgänge haben, während die anderen beiden umgekehrt eine Umsetzung der RS232-Pegel nach TTL-Norm vornehmen.

Damit läßt sich leicht eine normgerechte Schnittstelle nachrüsten. In Bild 2.69 sind sogar schon die entsprechenden Anschlüsse des C64-User-Port sowie die Pins am RS232-Stecker für eine bidirektionale Übertragung mit Handshake-Signalen angegeben.

Bild 2.70 zeigt das Layout für eine kleine Platine, die – ähnlich wie der Drucker-

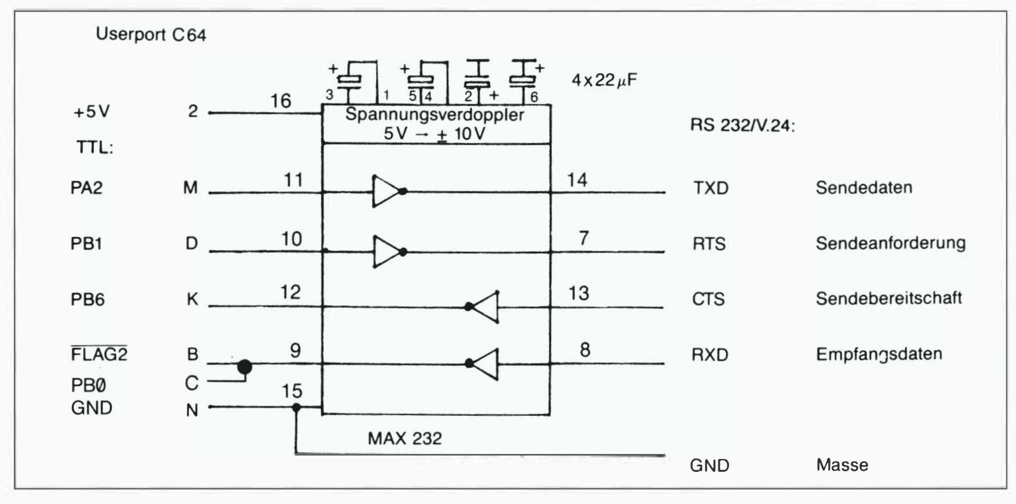


Bild 2.69: Schaltung der RS232-Platine und Blockschaltbild des MAX 232

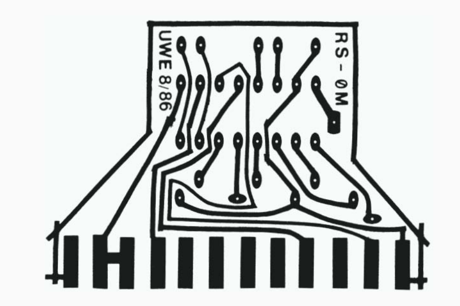
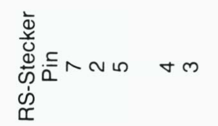


Bild 2.70: Eine Wandlerplatine für normgerechte RS232 Signale



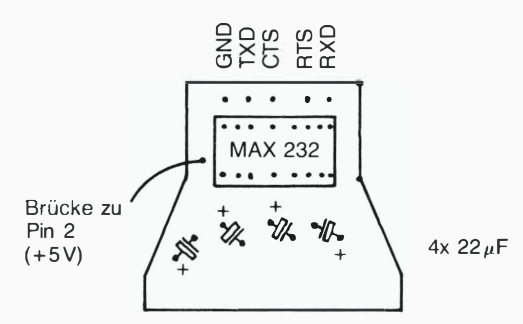


Bild 2.71: Bestückung der RS232-Platine

Treiber aus Kapitel 2.3.2 – in einem gewöhnlichen User-Port-Steckergehäuse Platz findet.

Den Bestückungsplan enthält Bild 2.71 und die benötigten Bauteile Tabelle 2.32.

1	Pegelwandler MAX 232 (erhältlich bei Völkner)
4	Elkos 22 Mikrofarad / 16 Volt, radial, kleine Bauform
1	User-Port-Stecker
1	User-Port-Steckergehäuse
1	einseitige Platine nach Bild 2.70

Tabelle 2.32: Bauteile für die RS232-Platine

Bitte richten Sie sich beim Aufbau wieder nach den Hinweisen im Anhang A. Auch bei dieser Platine darf jedoch das IC wegen der mangelnden Einbauhöhe nicht gesockelt werden. Aus demselben Grund ist bei den vier Elkos auf möglichst kleine Bauform zu achten. Sie sollten liegend in die Platine eingesetzt werden. Achten Sie auch auf die richtige Polarität.

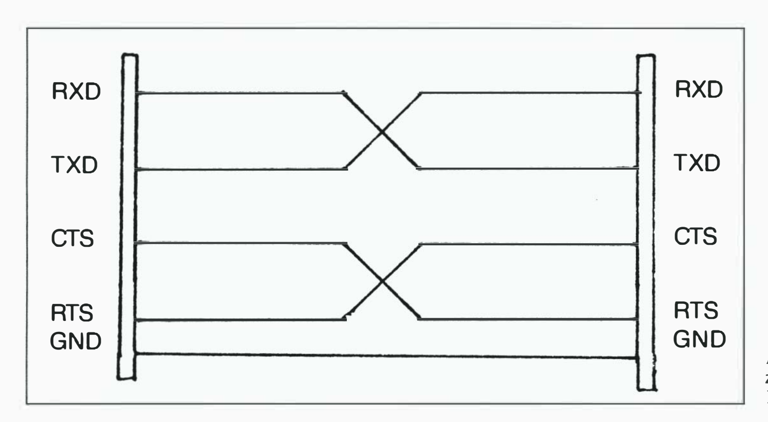


Bild 2.72: Zusammenschaltung zweier Platinen zu einem Nullmodem

Der User-Port-Stecker wird mit seiner unteren Kontaktleiste auf die vorgesehenen Kupferfahnen der Platine aufgelötet. Vergessen Sie nicht die Verbindung zu Pin 2 (+5V). Beim Einbau in das Gehäuse muß das RS232-Kabel mit dem Zugentlastungsbügel gut angeschraubt werden, um Beschädigungen durch ausgerissene oder gar sich berührende Kabelenden zu vermeiden.

Für echte RS232-Anwendungen sind Standardpegel unverzichtbar. So läßt sich der C64 mit Hilfe der vorgestellten Platine ohne weiteres als Terminal an leistungsstarke Computersysteme anschließen, wie zum Beispiel an den Amiga. Ebenso ist nun die Ansteuerung von Druckern mit serieller RS232-Schnittstelle gesichert.

Bild 2.72 zeigt, wie die RS232-Platinen zweier C64 miteinander zu verbinden sind, um über die Geräteadresse 2 Informationen austauschen zu können. Natürlich funktioniert ein solches »Nullmodem« auch ohne RS232-Platine, aber aus Kapitel 2.3.2.4 wissen Sie bereits, wie gefährlich das schon bei relativ kurzen Leitungen für den CIA-Baustein werden kann.

2.6.3 Fernschreiber am Computer

Fernschreiber waren die ersten Geräte, mit denen Informationen auf elektrischem Wege über weite Entfernungen transportiert werden konnten. Sie sind bis heute im Einsatz und haben die Geschicke vieler Entwicklungen mitbestimmt.

Auch in der Anfangszeit der elektronischen Datenverarbeitung stand der Fernschreiber als Ausgabemedium Pate. Die RS232-Schnittstelle entwickelte sich aus der Fernschreib-Norm. Damals benutzte man eine 20-mA-Stromschleife, die auch bei sehr langen Verbindungen weitgehend unempfindlich gegen beispielsweise elektromagnetisch eingekoppelte Störspannungen war, die sogenannte TTY-Schnittstelle (TTY = Teletype = Fernschreiber). Ein LOW-Signal kennzeichnet hier kein Stromfluß, während bei HIGH etwa 20 mA fließen.

Bei der TTY-Schnittstelle gibt es keine genormten Stecker und im Gegensatz zu Computerschnittstellen auch kein gemeinsames Bezugspotential (etwa Masse) zwischen Sender und Empfänger. Das ist ein Übrigbleibsel aus der früher bei Fern-

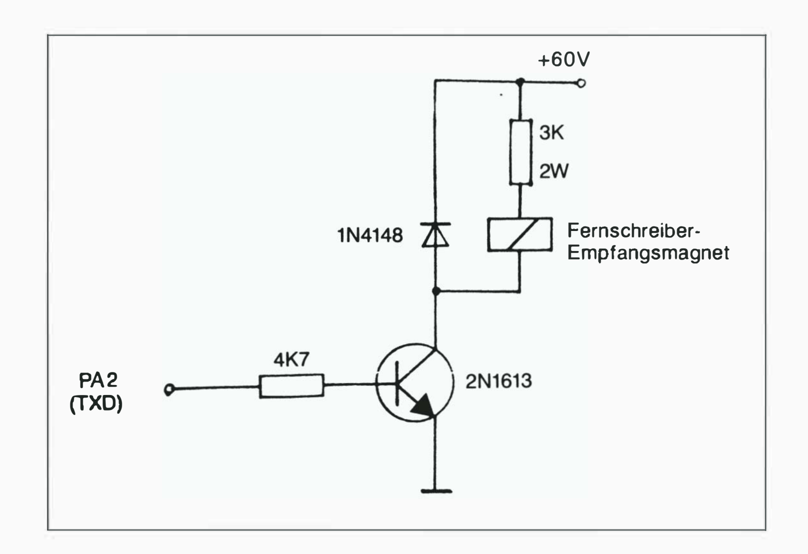


Bild 2.73: Ansteuerung des Empfangsmagneten eines Fernschreibers

schreibern üblichen Ansteuerschaltung. Bild 2.73 zeigt, wie ein solcher Fernschreiber als Drucker an den C 64 angeschlossen werden kann. Der Empfangsmagnet des Fernschreibers liegt zwischen der recht hohen Betriebsspannung von 60 bis 100 Volt und dem Kollektor eines Schalttransistors, an dessen Basis das Datensignal anliegt. Ein Kollektorwiderstand begrenzt den Strom auf etwa 20 mA, und eine Diode schützt den Transistor vor Beschädigung. Da die Betriebsspannung recht hoch ist, können auch Störspannungen von einigen Volt noch keine Übertragungsfehler verursachen.

Gängige Fernschreiber, die manchmal recht billig angeboten werden, sind die Geräte T37h oder T100 von Siemens. Um den Empfangsmagneten zu steuern, muß bei ihnen der braune und der gelbe Anschluß überbrückt werden. Die beiden anderen Drähte (grün und weiß) sind mit der Magnetspule verbunden. Hier sollte jedoch unbedingt die richtige Polarität ausgemessen werden.

Zur Ansteuerung von Fernschreibern muß das RS232-Ausgabeformat auf 5 Datenbit mit Start- und Stoppbits gestellt werden. Heute hat die TTY-Schnittstelle praktisch keine Bedeutung mehr, da sie keine genormte Steckerbelegung und keine Handshake-Leitungen besitzt. Sie wird nur noch in Spezialanwendungen eingesetzt. Da es zwischen den Geräten unterschiedliche Massepotentiale geben kann, werden meist Optokoppler zur galvanischen Trennung verwendet. Bild 2.74, Seite 154, zeigt eine entsprechende Schaltung.

2.6.4 Daten um die Welt

2.6.4.1 Zweitonmusik

Der wohl bekannteste Einsatzfall einer seriellen Datenübertragung ist die Benutzung des öffentlichen Telefonnetzes. Unzählige Fernsprecher auf der ganzen Welt lassen sich damit direkt anwählen. Nun darf aber nicht einfach die Telefonstrippe abgeschnitten werden, um sie an den User-

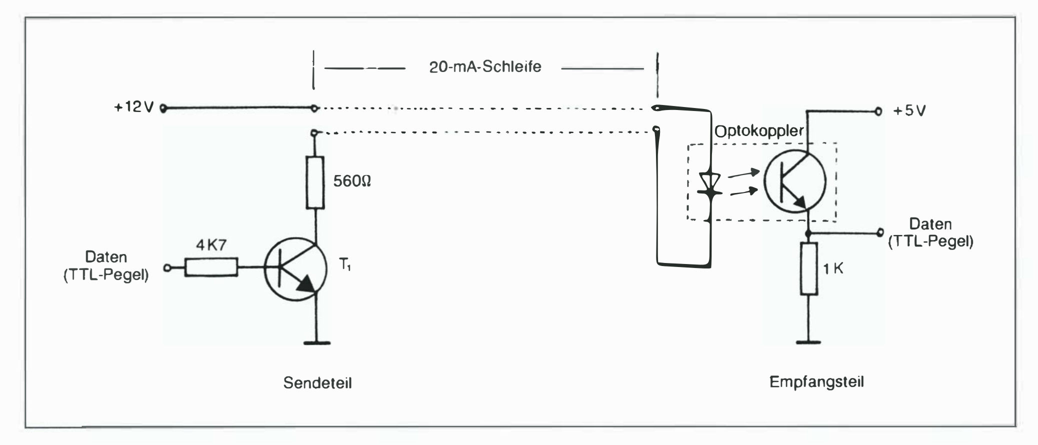


Bild 2.74: TTY-Schnittstelle mit galvanischer Trennung

Port anzuklemmen, denn zum einen ist das untersagt, und zum anderen würde eine solche Übertragung auch gar nicht funktionieren. Die Telefonleitungen sind für Tonfrequenzen gebaut – für den Wechselstrom also, der durch die Mikrofonkapsel beim Sprechen entsteht. Die Impulse unseres Computers aber sind eine Folge von Gleichströmen. Da im Telefonnetz an allen Ecken und Enden Transformatoren eingebaut sind, um bestimmte Abschnitte galvanisch voneinander zu trennen und weil alle paar Kilometer Tonfrequenzverstärker sitzen, die Leitungsverluste ausgleichen sollen, wäre ein solcher Versuch von vornherein zum Scheitern verurteilt.

Doch nicht gleich die Flinte ins Korn werfen! Der Gedanke liegt nahe, für die beiden Zustände HIGH und LOW einfach zwei unterschiedliche Tonfrequenzen einzusetzen, die – genau wie die Sprachfrequenzen – alle Hürden unbeschadet nehmen können. Selbst die Einspeisung ins Telefonnetz stellt kein Problem mehr dar, denn man kann ja einen Lautsprecher an den Hörer halten. Am anderen Ende werden diese Frequenzen mit einer ähnlichen Anordnung wieder in ihr logisches Äquivalent umgewandelt. Fachleute sprechen von Modulation und Demodulation der Daten. Ein Gerät, das die Umwandlung besorgt, heißt Modem (MOdulator/DEModulator) und wenn es die Tonfolgen über einen Lautsprecher zum Hörer übermittelt, ist es ein Akkustikkoppler.

Im dargestellten Fall lassen sich die Informationen nur in eine Richtung übertragen. Soll gleichzeitig empfangen und gesendet werden, dann wählt man einfach zwei weitere Frequenzen für den Rückkanal. Wegen des schon weiter oben vorgestellten Software-Protokolls zum Unterbrechen des Senders ist das wichtig, aber auch zur Eindämmung von Übertragungsfehlern. Jeder, der einmal über weite Strecken telefoniert hat, weiß, wieviel Knacken und Rauschen die Verständigung beeinträchtigen kann. Digitale Informationen sind da wesentlich empfindlicher als Sprache. Üblicherweise verfährt man deshalb folgendermaßen: Jedes übermit-

telte Zeichen wird vom Empfänger zurückgesendet und beim Sender mit dem Original verglichen. Nur wenn es korrekt war, wird es auf dem Bildschirm dargestellt. Andernfalls kommt ein Steuerzeichen und die Übermittlung wird wiederholt. Diese Übertragungsart nennt sich dann Vollduplex, bei einfacher Übermittlung – zwar in beiden Richtungen, aber ohne Kontrolle – handelt es sich um Halbduplex-Betrieb.
Natürlich (Wie sollte es anders sein?) sind die entsprechenden Frequenzen genormt, und ebenso selbstverständlich gibt es wieder abweichende Normen in verschiedenen Ländern. Damit Sie problemlos alle Datenbanken und Mailboxen der Welt ankurbeln können, hier die Bauanleitung für einen wirklich universellen Akkustikkoppler:

2.6.4.2 Ein Weltmodem

Das Herz des Zusatzes bildet ein spezieller Modem-Chip von Advanced Micro Devices, der nur wenige externe Bauelemente benötigt. Über Dip-Schalter kann er auf die verschiedenen Normen der Welt umgestellt werden. Für die Verbindung zum Rechner verfügt er über diverse TTL-Ein- und Ausgänge, von denen in unserem Fall allerdings nur TD (Transmit Data = Sen-

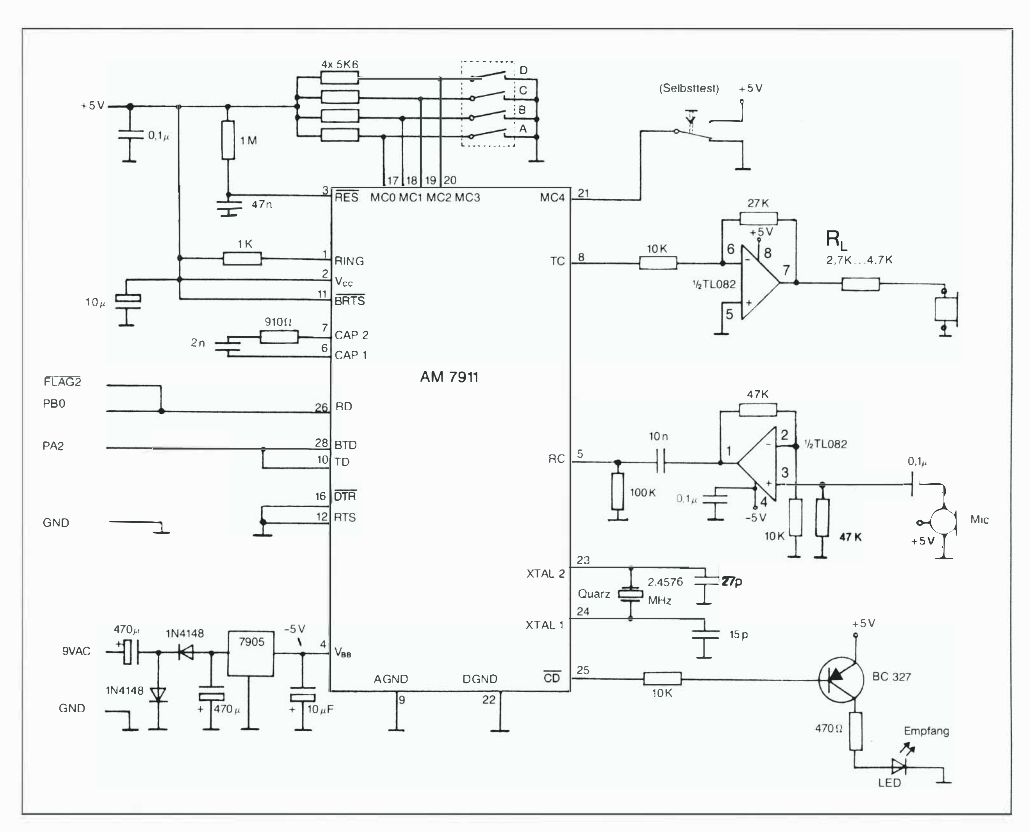


Bild 2.75: Schaltung des Weltmodems

dedaten) und RD (Received Data = Empfangsdaten) benutzt werden. Da auch der C64 nur über TTL-Pegel verfügt, wird eine Umwandlung auf RS232-Standard überflüssig.
Bild 2.75 zeigt den Schaltplan des C64-Weltmodems. Beinahe größeren Aufwand als das Modem selbst fordern der Schaltungsteil zur Erzeugung von –5 Volt aus der Wechselspannung am User-Port und die Verstärker zur akustischen Wandlung.
Einzelheiten über die Erzeugung der negativen Hilfsspannung lesen Sie bitte im Kapitel über Netzteilschaltungen (7.2) nach.
Zur Anpassung der analogen Signalamplituden dienen zwei Operationsverstärker (siehe Kapitel 2.2.11.1), die gemeinsam in einem IC vom Typ TL082 untergebracht sind. Dabei wird die Sendefrequenz vom Pin TC auf den einen Verstärker gebracht, so daß sie an dessen Ausgang über eine Telefonhörkapsel ausgegeben wird. Der andere Weg führt von einer Kondensatormikrofonkapsel, die in jedem guten Elektronikgeschäft für wenige Mark zu haben ist, auf den zweiten Verstärker und von dessen Ausgang über einen Kondensator zum Pin RC. Für den Verstärkungsfaktor ist jeweils der Gegenkopplungswiderstand vom nicht invertierenden Eingang zum Ausgang des Operationsverstärkers zuständig. Bei Bedarf kann er leicht geändert werden.

2.6.4.3 Aufbau des Modems

Bild 2.76 zeigt den Bestückungsplan für das Modem-Layout nach Bild 2.77. Richten Sie sich nach den Hinweisen im Anhang A. Der User-Port-Stecker wird mit

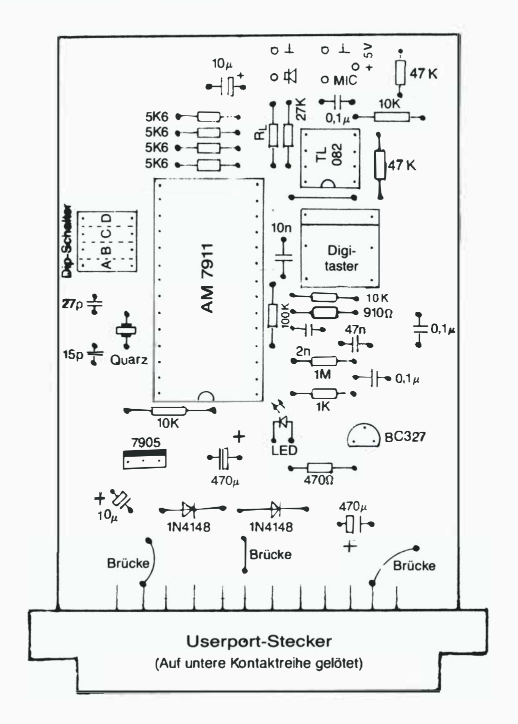


Bild 2.76: Bestückung des Weltmodems

seiner unteren Kontaktleiste auf die Platine gelötet und der Lötpunkt ganz rechts daneben mit Hilfe eines kurzen Drähtchens an Pin 2 auf der Oberseite des Steckers (+5V) kontaktiert. Weiterhin muß Pin 10 oder 11 des User-Port-Steckers (9 VAC) mit dem gekennzeichneten Punkt auf der Platine verbunden werden. Beim SX64 führt meist nur Pin 10 die entsprechende Spannung.
Achten Sie besonders auf die richtige Polung der Elkos und der Dioden. Wegen der negativen Hilfsspannung liegt der Elko hinter dem Spannungsregler mit seinem Pluspol an Masse!
Am hinteren Ende der Platine befinden sich die Anschlüsse für die Hörkapsel (links) und das Kondensatormikrofon (rechts). Der Pin ganz rechts führt

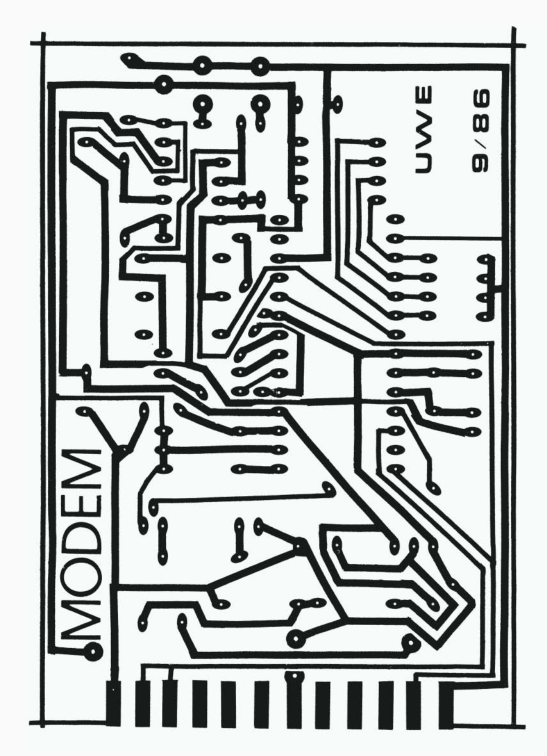


Bild 2.77: Modem-Layout

+5 Volt und kann zur Stromversorgung des Mikrofons dienen.
Akustikkoppler haben bei all ihrer Genialität einen gravierenden Nachteil: Sie sind von akustischen Umwelteinflüssen abhängig. Deshalb muß man Schallquelle und -aufnehmer so dicht wie möglich zusammenbringen und auf akustische Abschirmung gegen Umgebungsgeräusche Wert legen. Mikrofon als auch die Hörkapsel sollten in Gummimuffen eingebaut werden, die als Installationsbedarf für Sanitäranlagen in Baumärkten erhältlich sind. Foto 2.11 zeigt links eine geeignete Kondensatormikrofonkapsel, in der Mitte eine Telefonhörkapsel, die als Schallwandler geeignet ist, und rechts eine Gummimuffe mit eingebautem Mikrofon.

1 Modemchip AM 7911 (bzw. AM 7910)
1 IC-Sockel 28pol
1 Zweifach-Operationsverstärker TL082
1 IC-Sockel 8pol
1 Spannungsregler 7905
1 Transistor BC327 o.ä.
1 Leuchtdiode, Farbe nach Wahl
2 Universaldioden 1N4148
1 Quarz 2,4576MHz (HC-18/U)
1 Widerstand 1 Megaohm
1 Widerstand 100 Kiloohm
2 Widerstände 47 Kiloohm
1 Widerstand 27 Kiloohm
3 Widerstände 10 Kiloohm
4 Widerstände 5,6 Kiloohm
1 Widerstand 2,7...4,7 Kiloohm (R_L)
1 Widerstand 1 Kiloohm
1 Widerstand 910 Ohm (100 Ohm bei AM 7910)
1 Widerstand 470 Ohm
2 Elkos 470 Mikrofarad / 16 Volt, radial
2 Elkos 10 Mikrofarad / 16 Volt, radial
3 Kondensatoren 0,1 Mikrofarad, Keramik
1 Kondensator 47 Nanofarad
1 Kondensator 10 Nanofarad
1 Kondensator 2 Nanofarad
1 Kondensator 27 Pikofarad
1 Kondensator 15 Pikofarad
1 Dil-Schalter, 4fach
1 Digitaster, 1 x um
1 Kondensatormikrofonkapsel für 5-V-Betriebsspannung
1 Telefonhörkapsel
2 Installationsmuffen
1 User-Port-Stecker
1 einseitige Platine nach Bild 2.77

Tabelle 2.33: Die Bauteile des Weltmodems

Der Einbau des Modems in ein Gehäuse lohnt sich kaum. Wegen der fehlenden Zugentlastung für die Anschlußkabel sollten diese genügend lang sein. Für das Mikrofonkabel empfiehlt sich eine abgeschirmte Ausführung, wobei das Abschirmgeflecht mit dem hinteren Lötstift (Masse) verbunden werden sollte.

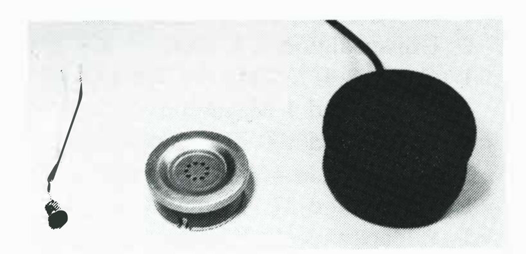


Foto 2.11: Elektroakustische Wandler

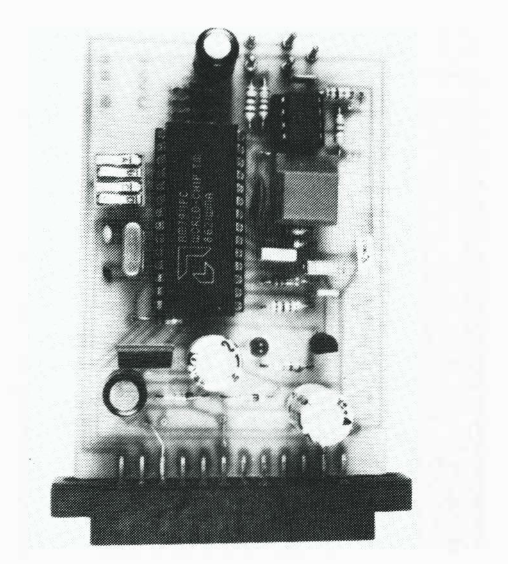


Foto 2.12: Fertiges Modem

Falls Sie den AM7911 nicht bekommen, können Sie auch den AM7910 verwenden, der lediglich einige Übertragungsnormen weniger beherrscht. In diesem Fall sollte aber der Widerstand an Pin 7 (910 Ohm) gegen einen mit 100 Ohm ausgetauscht werden. Foto 2.12 zeigt das fertig aufgebaute Modem. Zum Eigentest enthält die Platine einen Taster unterhalb des Operationsverstärkers. Stecken Sie die Schaltung dazu auf den User-Port und schalten den Rechner ein. Der Selbsttest erfolgt nach der Aufstellung von Tabelle 2.34 (Seite 159) dabei sind Dip-Schalter-Kombinationen einzustellen, die in Spalte ST mit »ja« gekennzeichnet sind. Wählen Sie ein geeignetes Übertragungsformat.
Nun laden Sie das kleine BASIC-Programm MODEMTEST und starten es mit RUN. Halten Sie den Taster gedrückt und betätigen Sie währenddessen einige Tasten des C64. Die zugehörigen Buchstaben müssen auf dem Bildschirm erscheinen. Damit haben Sie die Gewähr, daß zumindest der Modem-Chip richtig funktioniert.

2.6.4.4 Das Modem im Einsatz

Als Selbstbaumodem hat unser Gerät natürlich keine FTZ-Nummer und darf daher in der Bundesrepublik Deutschland nicht am öffentlichen Fernmeldenetz betrieben werden. Im Bundespostministerium wurde jedoch bereits eine deutliche Lockerung der Bestimmungen beschlossen, so daß in nächster Zeit mit der allgemeinen Betriebsgenehmigung für alle Akustikkoppler zu rechnen ist. Eine solche Regelung scheint vernünftig, denn ob nun ein Telefonteilnehmer in die Leitung zirpt, oder ein Gerät, bleibt sich doch gleich, solange nicht direkt in die Fernmeldeanlagen eingegriffen wird.
Es sollte selbstverständlich sein, daß unser Modem nur bei ausgeschaltetem Computer aufgesteckt werden darf. Danach ist es

bereits betriebsfertig. In den Rufnummernverzeichnissen der Mailboxen oder Datenbanken sind meist auch alle Übertragungsparameter angegeben. Stellen Sie die Dip-Schalter auf der Modem-Platine entsprechend ein. Tabelle 2.32 zeigt die verschiedenen Möglichkeiten.
Üblich ist CCITT V.21, Originate mit 300 Baud und Vollduplex. Dabei sind alle Schalter außer C geschlossen. Laden Sie ein Terminalprogramm. Dann stülpen Sie die Gummimuffe mit dem eingebauten Mikrofon auf die Hörseite und die mit der Hörkapsel auf die Sprechseite des Handapparates. Wählen Sie die Nummer der Gegenseite und warten auf das Herstellen der Verbindung. Hat alles geklappt, leuchtet die LED auf, zum Zeichen, daß eine Gegenstation empfangen wird. Jetzt kann der Datenaustausch losgehen.

D	C	B	A	ST	Übertragungsart
ein	ein	ein	ein	ja	Bell 103 Originate 300 Baud Vollduplex
ein	ein	ein	aus	ja	Bell 103 Answer 300 Baud Vollduplex
ein	ein	aus	ein	ja	Bell 202 1200 Baud Halbduplex mit 5 Baud Rückkanal
ein	ein	aus	aus	ja	Bell 202 1200 Baud Halbduplex mit Amplituden-Equalizer und 5 Baud Rückkanal
ein	aus	ein	ein	ja	CCITT V.21 Originate 300 Baud Vollduplex
ein	aus	ein	aus	ja	CCITT V.21 Answer 300 Baud Vollduplex
ein	aus	aus	ein	ja	CCITT V.23 Modus 2 1200 Baud Halbduplex
ein	aus	aus	aus	ja	CCITT V.23 Modus 2 mit Amplituden-Equalizer 1200 Baud Halbduplex
aus	ein	ein	ein	ja	CCITT V.23 Modus 1 600 Baud Halbduplex
aus	ein	aus	ein	ja	Bell 202 1200 Baud mit 150 Baud Rückkanal
aus	ein	aus	aus	–	Bell 202 1200 Baud mit Amplituden-Equalizer und 150 Baud Rückkanal
aus	aus	ein	ein	–	CCITT V.23 Mode 1 600 Baud mit Soft Turn-Off
aus	aus	aus	ein	–	CCITT V.23 Mode 2 1200 Baud mit Soft Turn-Off
aus	aus	aus	aus	–	CCITT V.23 Mode 2 1200 Baud mit Amplituden-Equalizer und Soft Turn-Off

Tabelle 2.34: Übertragungsnormen des AM7911

Kapitel 3
Der Kassetten-Port des C 64

Wie der kleine Bruder des User-Port mutet sein Nachbar an: der Kassetten-Port. Auch er bietet einige programmierbare Leitungen, die ganz und gar nicht allein dem Datenaustausch mit der Datasette vorbehalten bleiben brauchen.

3.1 Die Anschlüsse des Mini-Port

Anhang C enthält die Pinbelegung der Kassettenschnittstelle. Dargestellt ist der Blick von der Rückseite des Computers her auf die Kontakte. Der 12polige Platinendirektstecker täuscht eine Vielfalt von Signalen vor, doch in Wirklichkeit ist nur die Hälfte unterschiedlich belegt: jeweils übereinander liegende Kontakte sind kurzgeschlossen. Von den verbleibenden sechs Kontakten führt einer Massepotential, ein anderer +5 Volt, so daß vier nutzbare Steuerleitungen übrigbleiben.
Bild 3.1 zeigt die Ansteuerung der Kassettenschnittstelle vom C 64 aus. Verantwortlich für ihre Bedienung sind die Portleitungen P3, P4 und P5 der CPU 6510. Wie Sie wissen, enthält der Prozessor des C 64 einen 6-Bit-Port, der unter anderem auch die Speicheraufteilung im Bereich der Firmware zwischen ROM und RAM managt. Dieser Port funktioniert genauso, wie die beiden 8-Bit-Ports des 6526. Es gibt ebenfalls ein Datenrichtungsregister, das hier die Adresse 0 belegt, und die Portleitungen als Eingang bzw. Ausgang definiert. Zum Festlegen der Ausgabepegel bzw. zum Lesen der Eingangspegel dient Adresse 1. Einziger Unterschied zu den Ports der CIAs ist, daß die beiden Bits 6 und 7 beim CPU-Port nicht herausgeführt sind. Die zwischengeschalteten kleinen Induktivitäten (FB) sind Ferritperlen, die Störeinflüsse auf der Leitung vermindern sollen. Wir sehen, daß die Portleitung P3 direkt an den Stecker geführt ist. Sie ist im normalen Betrieb als Ausgang programmiert und überträgt die Daten zum Schreiben auf das Band. Dagegen ist die Portleitung P4 ein Eingang. Es handelt sich um den Anschluß »SENSE«, der im Innern der Datasette bei gedrückter PLAY-Taste auf Masse gelegt wird. Der nötige Pull-Up-Widerstand (siehe Kapitel 2.2.4.2) sitzt als R_1 im Computer.
Daß P5 eine Sonderfunktion erfüllt, ist auf den ersten Blick klar. Der Schaltungsteil mit den vielen Halbleitern stellt eine abschaltbare Spannungsstabilisierung dar, mit der einfach die Motorspannung der Datasette aus- und eingeschaltet werden kann.

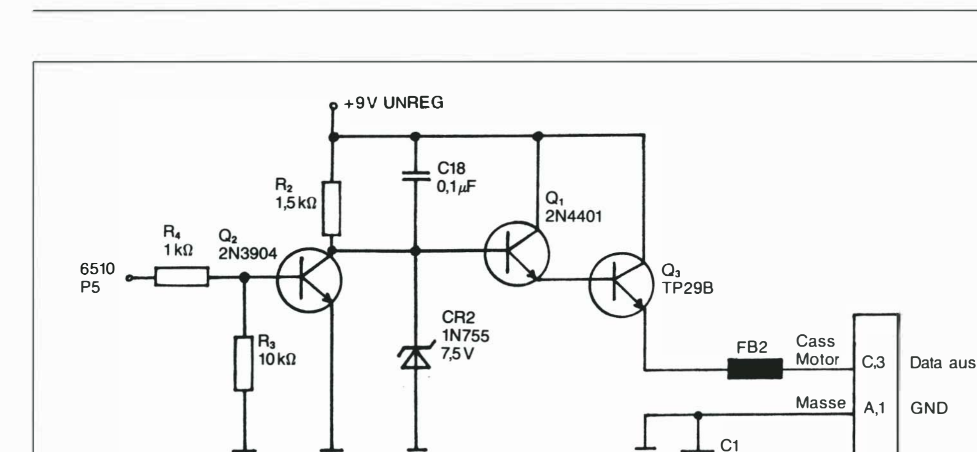


Bild 3.1: Ausschnitt aus dem C64-Schaltplan

Für das Lesen der Daten vom Band schließlich wurde kein Portanschluß gewählt, sondern der Eingang $\overline{\text{FLAG}}$ des CIA # 1 findet dafür Verwendung. Die Vorteile dieses Anschlusses wurden bereits in Kapitel 2.3 erläutert. Hier ist sein Einsatz sinnvoll, da bei magnetischer Aufzeichnung die Informationen immer im Magnetisierungswechsel stecken.

3.2 Das Commodore-Kassettenaufzeichnungsformat

Commodore wählte beim Aufzeichnungsformat auf Kassette einen unüblichen Weg. Es wird mit insgesamt drei unterschiedlichen Frequenzen gearbeitet, die mit »Byte«, »Long« und »Short« bezeichnet werden. Die Kennzeichnung eines Byteanfangs geschieht nicht – wie sonst üblich – durch ein Startbit, sondern mit der besonders reservierten Frequenzkombination Byte-Long. Daher ist auch kein Stoppbit nach jedem Byte erforderlich, allerdings wird statt dessen ein neuntes Bit zur Paritätsprüfung angehängt.

Auf dem Band wird in diesem Format beim Speichern eines Programms nacheinander folgendes aufgezeichnet:

Zunächst 10 Sekunden Vorspann, der dafür sorgt, daß bei zurückgespulter Kassette trotz Vorspannband nichts verlorengeht.

Der darauffolgende Header-Block enthält folgende Informationen:

1. 9 Synchronisationsbyte ($89..81)
2. Flag-Byte: 01 = BASIC-Programm
   02 = Maschinenprogramm
   04 = Daten
3. 4 Byte mit Anfangs- und Endadresse
4. Filename (mit Leerräumen auf 128 Byte aufgefüllt)
5. Prüfsummen-Byte

Für den Fall, daß der Header beim erstenmal nicht korrekt gelesen werden konnte, wird er insgesamt nach einer 50-ms-Pause noch einmal wiederholt.
Der Programmblock bzw. Datenfileblock ist wie folgt aufgebaut:

1. 9 Synchronisationsbyte ($89..81)
2. Datenbytes (codiertes BASIC-Programm, nur Textfiles in reinem ASCII)
3. Prüfsumme

Auch der Programmblock wird zur Sicherheit zweimal nacheinander aufgezeichnet. Obwohl sich physikalisch eine Geschwindigkeit von knapp 1200 Baud ergibt, sind wegen der zweimaligen Aufzeichnung mit nur 60 Byte pro Sekunde zu rechnen. Das entspricht 600 Bit pro Sekunde, da jedes Byte aus zehn Zellen besteht.

3.3 Normaler Kassettenrecorder statt Datasette

Der Kassettenanschluß des C64 ist nicht für normale Kassettenrecorder vorgesehen, sondern für eine spezielle Sonderversion mit TTL-Ein- und Ausgangspegeln. Mit der Schaltung nach Bild 3.2, Seite 164, können jedoch auch handelsübliche Recorder verwendet werden. Die Aufzeichnung der Daten macht dabei keinerlei Schwierigkeiten. Ein einziger Widerstand genügt, um die Ausgangsspannung an die Eingangsempfindlichkeit eines Mikrofoneingangs anzupassen.
Komplizierter wird dagegen die Aufbereitung der vom Recorder gelieferten Informationen. Wie bereits in Kapitel 2.2.4.6 erläutert, arbeiten auch hier gleich mehrere Gatter eines CMOS-Negierer-IC 4069 als Verstärkerstufen. Der Eingang der Schaltung wird an den Ohrhörer- oder Lautsprecherausgang des Recorders geschaltet. Je nachdem, ob der verwendete Recorder die Ausgangsspannung invertiert oder nicht, muß der Polaritätsschalter in die entsprechende Stellung gebracht werden. Die folgenden parallelgeschalteten Gatter (Bei CMOS ist das möglich!) erzeugen schließlich das vom Rechner erwartete TTL-Signal.
Der untere Teil des Schaltplans muß nicht unbedingt aufgebaut werden. Er besorgt die von der Datasette bekannte Motorsteuerung. Dazu wird mittels eines Relais die Stromzuführung des Motors unterbrochen. Oft haben Kassettenrecorder Anschlüsse für ein Mikrofon mit START/STOP-Schalter, der über einen zusätzlichen 2,5-mm-Klinkenstecker ange-

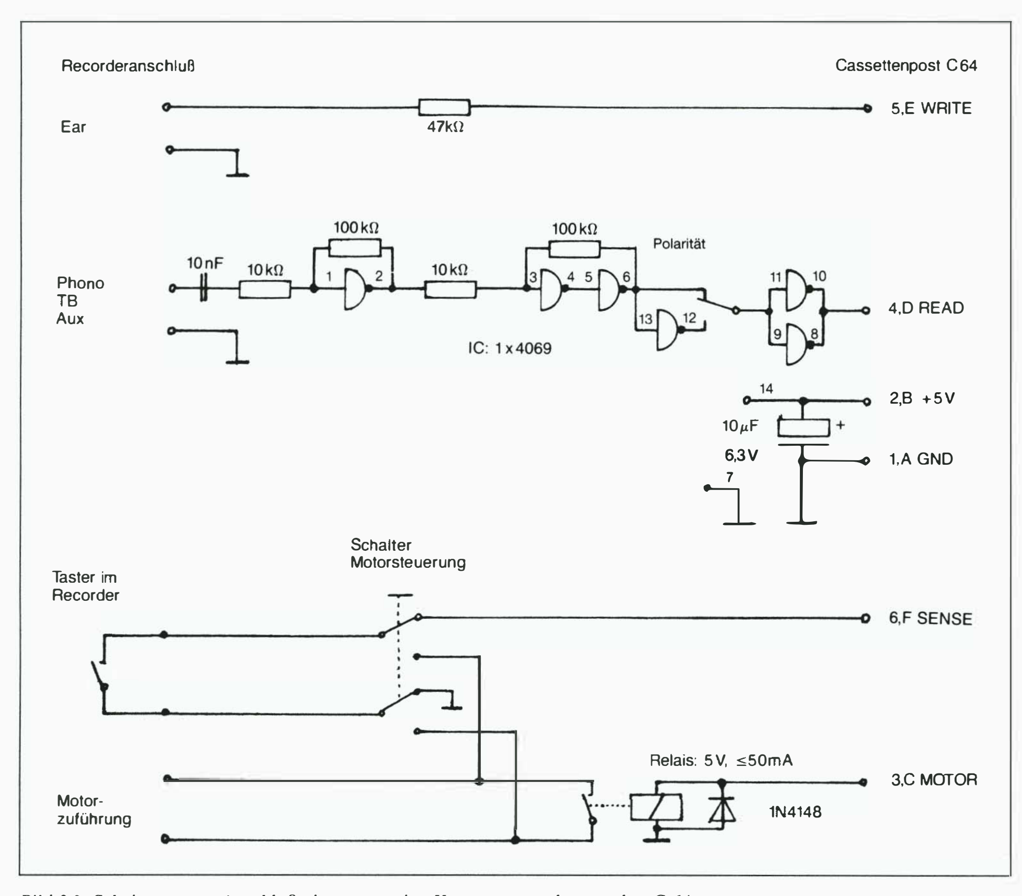


Bild 3.2: Schaltung zum Anschluß eines normalen Kassettenrecorders an den C 64

schlossen wird. Genau über diesen Stecker läßt sich die Motorsteuerung auch ohne Eingriffe in den Recorder realisieren.

Nicht möglich ist das allerdings bei dem »SENSE«-Taster. Er ist an geeigneter Stelle in der Mechanik einzubauen. Eventuell können Sie auch einen bereits eingebauten Kontakt mitbenutzen. Er muß nur auf einer Seite mit Masse verbunden sein.

Um den Aufwand zu verringern, kann auch ein zusätzlicher Schalter eingebaut werden, der dann gleichzeitig mit der Play-Taste zu bedienen ist.

3.4 Ein I²C-Bus am C 64

Um den Kassetten-Port nur zum Anschluß eines Recorders zu verwenden, ist er viel zu schade. Der folgende Abschnitt zeigt eine ganz andere Verwendungsmöglichkeit. Es soll ein System vorgestellt werden, mit dem sich die Vorteile spezieller ICs auch von Nichtfachleuten voll nutzen lassen, obwohl sie nur mit ein paar Drähtchen am Kassetten-Port angeschlossen werden.

3.4.1 Was ist der I²C-Bus?

Die bisherigen Schnittstellen waren ausschließlich dazu bestimmt, genau zwei – und nicht mehr – Geräte miteinander zu verbinden. Der Ausdruck »Bus« bezeichnet in der Computertechnik grundsätzlich ein Leitungssystem zum Austausch von Informationen zwischen mehreren Sendern und Empfängern. Es gibt viele unterschiedliche Realisierungsmöglichkeiten für diese Verkehrswege. Jede hat dabei Vor- und Nachteile. Bekannte genormte Bussysteme sind etwa der serielle IEC-Bus, ähnlich wie er beim Anschluß der Commodore-Floppy VC 1541 an den C64 angewandt wird, oder der in professionellen Steuersystemen oft eingesetzte parallele VME-Bus. Bei einer parallelen Lösung werden viele Daten gleichzeitig übertragen, während sie bei einem seriellen Konzept zuerst aufgeteilt werden und dann schön nacheinander über das Kabel gehen müssen. Eine parallele Übertragung kann wesentlich schneller sein. Dem steht ein deutlich geringerer Hardwareaufwand bei der seriellen Übermittlung gegenüber. Nicht nur das Kabel allein ist hier maßgebend (viele Adern können bei langen Strecken schon recht teuer sein), sondern vor allem die Geräte selbst kommen mit weniger Aufwand aus. Außer den üblichen Steueradern wird nur noch eine einzige Datenleitung benötigt. Das wirkt sich auf die Pinzahl und damit die Gehäusegröße der eingesetzten ICs aus. Bei kleineren Komponenten läßt sich auch die benötigte Platinenfläche der Schaltung verkleinern, die einen beträchtlichen Anteil an den Gesamtkosten eines Systems ausmacht.
Die Stromaufnahme der Schaltung sinkt, unter Umständen kann das Netzteil kleiner werden und eventuell sogar das ganze Gerät. Auf jeden Fall wird die Herstellung billiger.
Außerdem hat die Erfahrung gezeigt, daß die Störanfälligkeit eines Systems mit der Anzahl der Steckverbindungen wächst.
Der I²C-Bus ist eine Minimalschnittstelle von der Firma Valvo. Die im folgenden beschriebene Routine I²C-BUS-HANDLING ermöglicht es, Bausteine mit dieser Schnittstelle auch am C64 zu betreiben. Das eröffnet viele Chancen, einerseits das breite Angebot an interessanten Bausteinen mit I²C-Schnittstelle selbst zu nutzen, und andererseits mit dem Heimcomputer einmal in ein kommerzielles Gerät hineinzuschnuppern, und eventuell sogar dessen Steuerung von einem selbstgeschriebenen (BASIC-)Programm übernehmen zu lassen!
Als Beispiel soll hier die Ansteuerung einer Uhren-/Kalenderschaltung dienen, mit der ständig, auch wenn der Computer zwischendurch mal abgeschaltet wurde, die aktuelle Uhrzeit zur Verfügung steht.

3.4.2 Wie funktioniert der I²C-Bus?

Der I²C-Bus ist für den bidirektionalen 2-Draht-Datenverkehr zwischen verschiedenen integrierten Schaltungen oder Modulen ausgelegt. Die beiden Leitungen sind eine serielle Datenleitung (SDA) und eine serielle Taktleitung (SCL). Bild 3.3 zeigt die prinzipielle Zusammenschaltung von verschiedenen Teilnehmern. Einen Baustein, der am I²C-Bus eine Information erzeugt und aussendet, nennt man »Sender«; eine Einheit, die eine Information entgegennimmt, ist ein »Empfänger«.

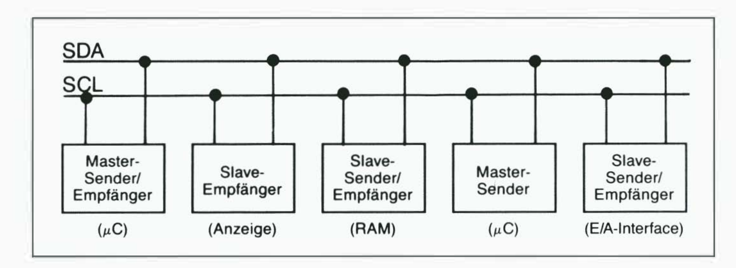


Bild 3.3: Verschiedene Bausteine am I²C-Bus

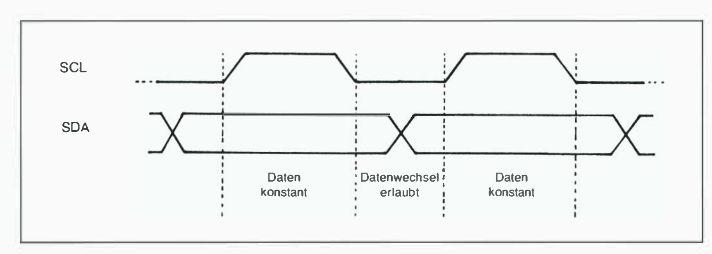


Bild 3.4: Abhängigkeit der Datenleitung SDA von der Taktleitung SCL

Die Schaltung, die eine Übertragung von Informationen steuert, wird als »Master« bezeichnet, die vom Master gesteuerten Schaltungen heißen »Slaves«. Der Master generiert ein Taktsignal auf der Clock-Leitung SCL. Während eines jeden Taktimpulses wird ein Datenbit übertragen. Bild 3.4 macht das deutlich.
Ein Datenwechsel darf nur während der LOW-Phase des Taktsignals erfolgen. Die einzige Abweichung von dieser Regel bildet der Anfang einer Übertragung. Ein Wechsel der Datenleitung von HIGH nach LOW, während die Taktleitung HIGH-Potential führt, gilt als Startbedingung und leitet jede Busübertragung ein.
Es können durchaus mehrere Master in einem System existieren, es darf aber nur immer einer davon den Bus belegen. Ein Slave-Baustein kann Sender sein, wenn er von einem Master angesprochen wurde und Antwort gibt. Nach wie vor kommt jedoch das Taktsignal dann vom jeweiligen Master. Bei nicht belegtem Bus verbleiben sowohl die Daten- als auch die Taktleitung im HIGH-Zustand.
Die Anzahl der bei einem Buszugriff übertragbaren Bytes ist nicht begrenzt. Jedes

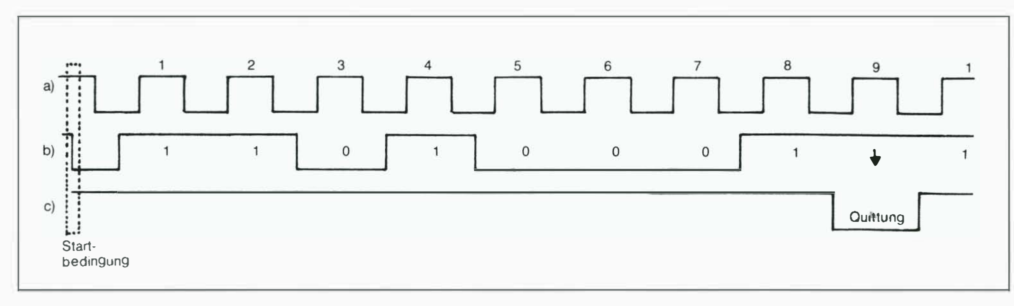


Bild 3.5: Beispiel einer Datenübertragung (Adreßbyte lesen)
a) Taktsignal SCL b) Daten SDA vom Sender (Computer) c) Datenleitung vom Empfänger (PCB 8573)

Byte besteht aus 8 Bit, denen ein Quittierbit (Acknowledge) folgt. Bild 3.5 zeigt die genauen Abläufe bei der Übertragung des ersten Byte. Zuerst erzeugt der Sender eine Startbedingung. Die Taktleitung geht daraufhin auf LOW und die Spannung für das erste Bit kann auf die Datenleitung gelegt werden. Immer wenn die Clock-Leitung HIGH ist, wird die Information vom Empfänger übernommen. In unserem Beispiel entsteht die Bitfolge 11010001. Das ist die Adresse des Uhren-/ Kalenderbausteins PCB 8573 zum Lesen. Ihre Bedeutung wird später noch ausführlich erläutert.

Nach der Übermittlung des achten Datenbit ändert sich die Datenrichtung kurz. Der Sender beläßt die Datenleitung auf HIGH. Dagegen zieht nun der Empfänger diese Leitung während des nächsten Taktimpulses auf LOW. Damit quittiert er dem Sender, daß die Information angekommen ist. Durch einen fehlenden Quittungsimpuls wird das Ende der Datenübertragung signalisiert. Der Master sendet in diesem Fall eine Stoppbedingung aus, indem er – ähnlich wie bei der Startbedingung – die Datenleitung von LOW nach HIGH gehen läßt, während die Taktleitung konstant HIGH bleibt.

Die maximale Taktfequenz des I^2C-Bus beträgt 100 kHz.

Bevor die Datenübermittlung auf dem I^2C-Bus beginnen kann, muß der Master nachsehen, ob der Bus nicht schon belegt ist, um nicht in eine bereits laufende Übertragung hineinzuplatzen. Sollte alles klar sein, wird zunächst die gewünschte Schaltung mit dem ersten Byte nach der Startbedingung adressiert. Ist sie ansprechbar, reagiert sie darauf mit einem Quittungsbit, und die Datenübertragung kann beginnen.

Die genannten Abläufe werden alle von der Handling-Routine erledigt. Wie der Betrieb mit den angeschlossenen Schaltungen genau abläuft, soll später am Beispiel der Uhren-/Kalenderschaltung noch deutlich gemacht werden.

3.4.3 Die I^2C-Schnittstellen-Hardware

Um eine I^2C-Schnittstelle am C64 zu installieren, wurde der Kassetten-Port gewählt. Alle anderen programmierbaren Leitungen können bereits von unverzichtbaren Geräten belegt sein, wie zum Beispiel der User-Port von einem Centronics-Drucker. Allein die Kassettenschnittstelle bleibt meist frei, zumal doch die Mehrzahl der ernsthaften Anwender überhaupt ausschließlich mit der Floppy am Computer arbeitet. Außerdem steht hier genau die benötigte Anzahl von Leitungen zur Verfügung. Daher wird nichts überflüssig verbaut und verschwendet. Bei vielen Schnelladesystemen für die Floppy VC 1541 sind ohnehin alle Kassettenroutinen entfernt, so daß dieser Port nicht mehr für seinen ursprünglichen Zweck genutzt werden kann. Das stört die hier vorgestellte I^2C-Routine aber keineswegs.

Schauen Sie sich noch einmal den Ausschnitt aus dem Schaltplan des C64 in Bild 3.1 an. Für die Steuerung der Clock-Leitung SCL wurde der Anschluß CASS WRT gewählt. Diese Leitung muß am Computer als Ausgang definiert sein. Über sie überträgt der C64 im normalen Betrieb die Informationen zum Schreiben auf Band an die Datasette und ist daher ebenfalls gewöhnlich als Ausgang pro-

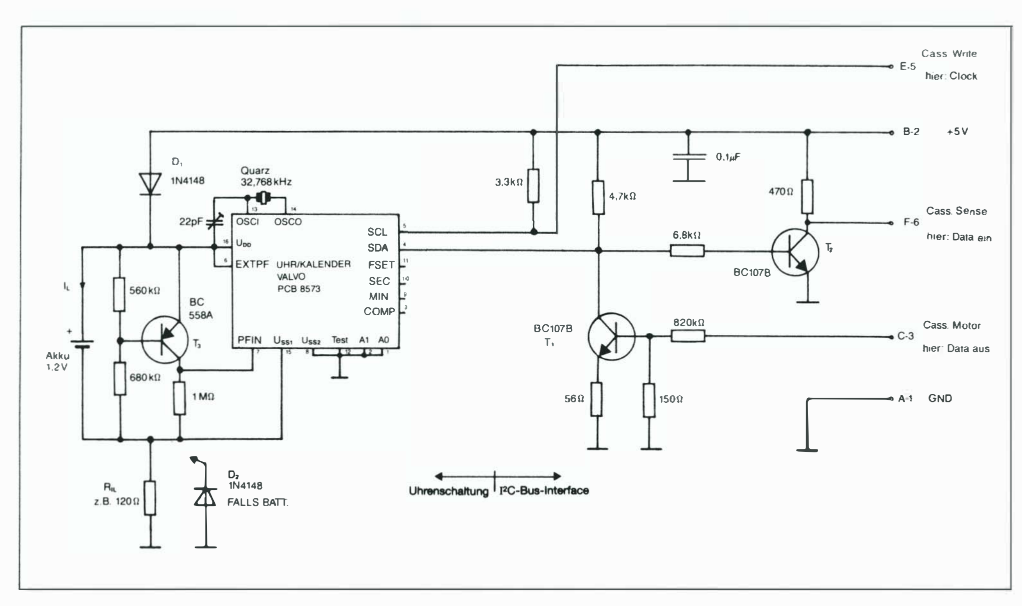


Bild 3.6: Der Schaltplan des I²C-Interface (rechts) und der daran angeschlossenen Uhrenschaltung

grammiert. Sollte bei einem Aufruf der Handling-Routine versehentlich die Datasette angeschlossen sein, können so keine Hardwareschäden dadurch entstehen, daß zwei Ausgänge direkt zusammengeschaltet sind und eventuell gegeneinander arbeiten.

Leider läßt sich die bidirektionale Datenleitung der I²C-Bausteine nicht ebenfalls unmittelbar mit einer normalen Portleitung ansteuern, da hierfür im Computer Open-Collector-Ausgänge vorliegen müßten. Es war daher eine Aufteilung in Ausgang DATAAUS und Eingang DATAEIN nötig. Diese Trennung besorgt die kleine I²C-Interfaceschaltung, die in der rechten Hälfte von Bild 3.6 zu sehen ist. Gleichzeitig wird dort auf der linken Seite die Anschaltung des Uhren-/Kalenderbausteins PCB 8573 mit seinen externen Bauelementen dargestellt.

Der Transistor T_1 bildet die Sendestufe. Er wird von der Motorspannung am Steckerpin »C,3« angesteuert. Ein HIGH-Pegel an dieser Stelle läßt den Transistor T_1 durchschalten und zieht die Datenleitung SDA auf LOW-Potential. Wie in Bild 3.1 zu sehen ist, wird die Motorspannung im Inneren des C64 mit Hilfe des Transistors Q_2 gesteuert. HIGH-Pegel am Portanschluß P5 des 6510 schaltet diesen Transistor durch, so daß die Basisspannung von Q_1 etwa 0 Volt beträgt. Q_3 sperrt damit, und über den Schnittstellenanschluß »C,3« liegt LOW-Pegel an der Basis von T_1 der Interfaceschaltung an: SDA bleibt unbeeinflußt HIGH. Erst eine 0 am Port-Anschluß P5 des Prozessors schaltet T_1 durch, und SDA wird ebenfalls 0.

Probleme bei dieser Beschaltung bereitet lediglich der Kondensator C_{18} im Inneren des C64. Er soll bei der ursprünglichen Verwendung zur Glättung der geregelten Motorspannung beitragen, da er Span-

nungsschwankungen mit Hilfe seiner Speicherwirkung auffängt. Bei einem Spannungswechsel verzögert dieser Kondensator die Durchschaltung um einen gewissen Wert. Genau das ist aber beim schnellen Übertragen von Daten, also bei kurz aufeinanderfolgenden Zustandswechseln der Schaltspannung äußerst unerwünscht. Da der Kondensator glücklicherweise nur einen relativ kleinen Kapazitätswert aufweist (0,1μF), konnte der störende Effekt durch eine Spannungsteilerschaltung am Basisanschluß des Transistors T_1, die den Reglerausgang »C,3:« relativ niederohmig belastet, weitgehend eingedämmt werden. Im Handling-Programm ist eine Zeitschleife (LOOP) eingebaut, die nach jedem Pegelwechsel einige Mikrosekunden wartet, bis der Spannungswert stabil ist.

Änderungen im Computer werden nicht nötig.

Der zweite Transistor T_2 ist für die entgegengesetzte Datenrichtung zuständig. Er dient lediglich zur Entkopplung gegen Rückwirkungen vom Computeranschluß »F,6«. Hier gilt es jedoch zu beachten, daß die Transistorstufe invertierend wirkt. HIGH-Pegel auf der Datenleitung SDA wird als logische 0 am Portanschluß P4 des 6510 gemeldet und umgekehrt.

3.4.4 Aufbau und Inbetriebnahme der Zusatzschaltung

Die kleine Zusatzschaltung kann auf einer Platine nach Bild 3.7 aufgebaut werden. Der Bestückungsplan (Bild 3.8) zeigt die genaue Lage aller Bauelemente aus der Liste (Tabelle 3.1). Alle benötigten Teile sind handelsüblich. Falls die Beschaf-

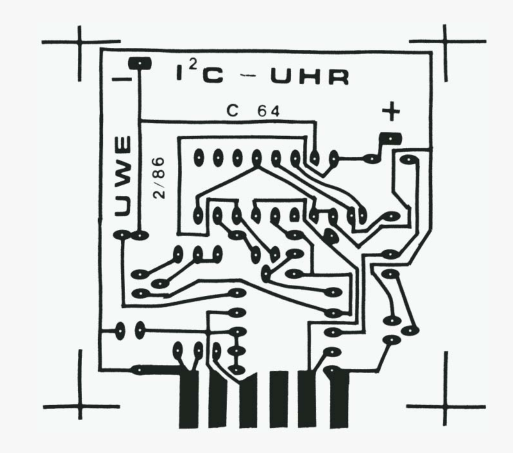


Bild 3.7: Layout für das I²C-Interface mit Uhr und Kalender

1	PCB 8573 (= PCF 8573) (ca. DM 8,-)
2	Transistoren BC 107 B o.ä.
1	Transistor BC 558 A o.ä.
1	Diode 1N4148 o.ä.
1	Widerstand 1 Megaohm
1	Widerstand 680 Kiloohm
1	Widerstand 560 Kiloohm
1	Widerstand 6,8 Kiloohm
1	Widerstand 4,7 Kiloohm
1	Widerstand 3,3 Kiloohm
1	Widerstand 820 Ohm
1	Widerstand 470 Ohm
1	Widerstand 150 Ohm
1	Widerstand 120 Ohm
1	Widerstand 56 Ohm
1	Kondensator 0,1 Mikrofarad
1	Trimmkondensator 22 Pikofarad
1	Miniaturquarz 32,768KHz
1	Kassetten-Port-Stecker für C64 (z.B. TRW 252-06-50-100)
1	IC-Sockel 16polig
1	einseitige Platine nach Bild 3.7

Tabelle 3.1: Stückliste für I²C-Interface und Uhr

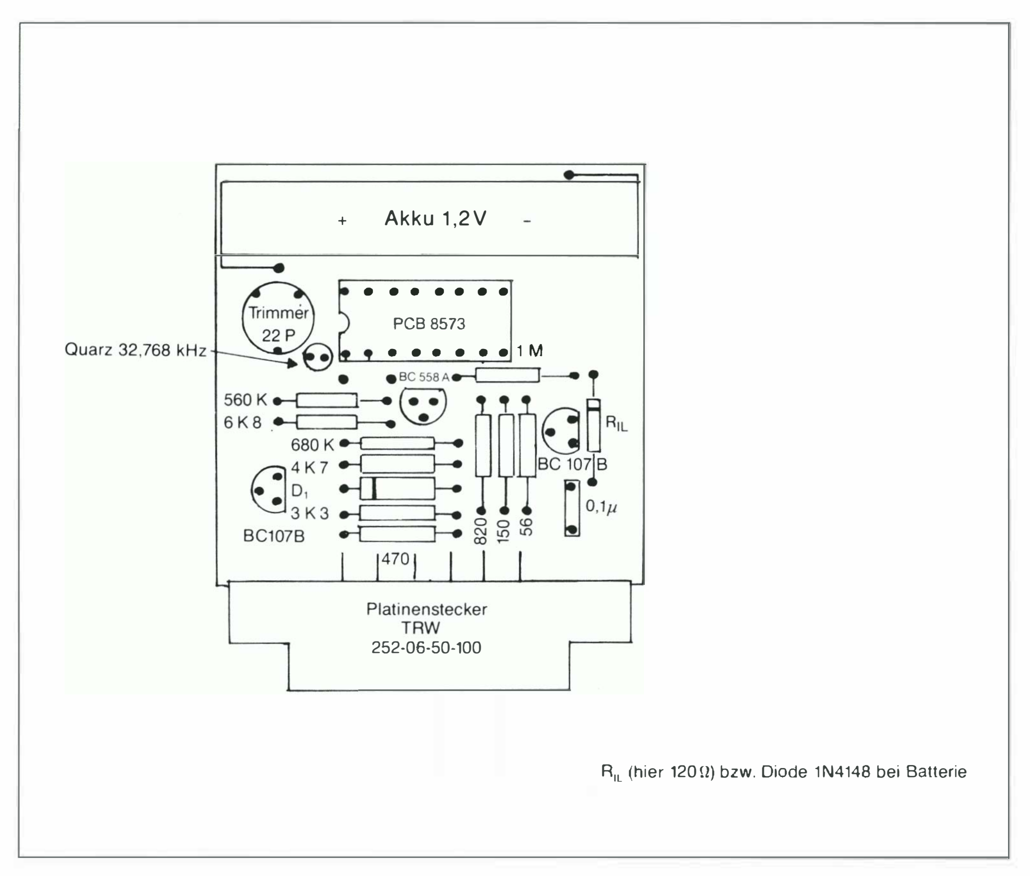


Bild 3.8: Bestückungsplan für I²C-Interface und Uhr

fung des Uhrenbausteins Schwierigkeiten macht, wenden Sie sich an die Firma Schmotz Electronik , Postfach 14 12, 8202 Bad Aibling.

Die fertig aufgebaute Platine wird mit den Lötfahnen auf der unteren Kontaktleiste des Kassetten-Port-Steckers befestigt. Verbindungen zur oberen Leiste sind nicht nötig, da die beiden übereinander befindlichen Kontakte jeweils schon intern verbunden sind.

Vorsicht ist beim Anlöten des Akkus geboten. Seine Kontakte dürfen nicht überhitzt werden und natürlich ist auf die richtige Polung zu achten. Falls eine Batterie zum Einsatz kommen soll, ist der Ladewiderstand R_{IL} durch eine Diode zu ersetzen. Andernfalls richtet sich sein Wert nach dem gewünschten Ladestrom. Der Akku wird bei einem Widerstandswert von 120 Ohm mit einem Strom von etwa 20 mA geladen, solange der Rechner eingeschaltet ist.

Haben Sie Ihren Computer täglich sehr lange in Betrieb, ist ein kleinerer Ladestrom und damit ein größerer Widerstandswert von etwa 1 Kiloohm empfehlenswert.

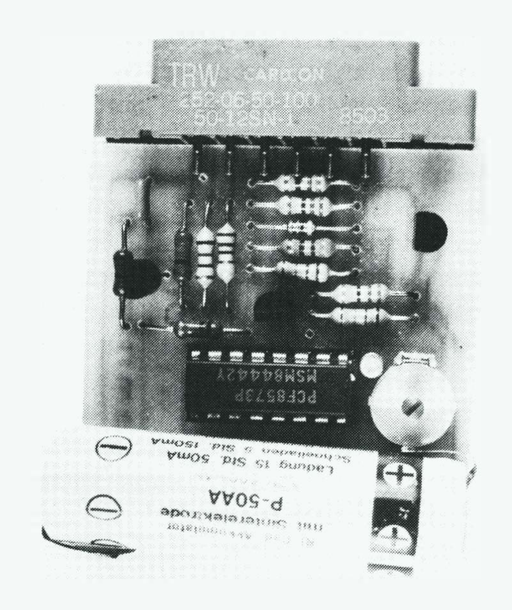


Foto 3.1: Das betriebsbereite I^2C-Zusatz-Gerät

Die Diode D_1 verhindert einen Stromfluß vom Akku durch den Rechner und damit eine zu schnelle Entladung im Ruhebetrieb.

Das Uhren-IC sollte, schon wegen der thermischen Belastung durch das Einlöten, gesockelt werden.

Foto 3.1 zeigt das betriebsbereite Mustergerät.

Nach dem Zusammenbau folgt der Abgleich. Jeder Quarz hat eine gewisse Toleranz. Beim Mustergerät war ohne jeglichen Abgleich über mehrere Wochen keine merkbare Abweichung festzustellen, doch um die Uhr wirklich haargenau gehen zu lassen, kann mit dem Drehkondensator ein Feinabgleich vorgenommen werden. Dazu steht am Anschluß FSET (Pin 11) des Uhrenbausteins das frequenzmäßig durch 256 geteilte Oszillatorsignal rückwirkungsfrei zur Verfügung. Es kann mit einem präzisen Frequenzzähler (zum Beispiel dem C64-Frequenzmeter) auf exakt 128 Hz abgeglichen werden.

Stecken Sie dazu (bei ausgeschaltetem Rechner!) die Frequenzmeter-Platine aus Kapitel 2 auf den User-Port, während sich die I2C-Platine am Kassetten-Port befindet. Verbinden Sie den linken Eingangsstift mit Pin 11 des Uhren-IC PCB 8573 auf der I^2C-Platine und starten Sie das BASIC-Programm FREQUENZMETER. Die Anzeige muß um 128 Hz schwanken. Gleichen Sie mit dem Trimmkondensator auf exakt 128 Hz ab. Für verläßliche Ergebnisse müssen natürlich die Torzeitwerte im Meßprogramm auf Ihren Computer eingestellt sein. Genauere Angaben finden Sie im Kapitel 2.4.4.

Ebenso ist es auch möglich, den Baustein über eine lange Zeit hinweg zu kontrollieren, und durch Versuche die beste Einstellung herauszufinden.

3.4.5 Der Einsatz der I^2C-Handling-Routine

Das Maschinenprogramm I2C-HANDLING macht den C64 mit der Interfaceschaltung zu einem Master Sender/Empfänger am I^2C-Bus. Es belegt den Speicherbereich \$C000 bis \$C1B2 (dezimal: 49 152 bis 49 586). Der Hauptteil erledigt die Abfrage des Bus vor einem Zugriff, die Erzeugung der Start- und Stoppbedingungen und das Senden von Daten. Falls eine Antwort erwartet wurde, verzweigt er zum Empfangsteil. Das Wechseln der Busleitungen wird mittels Unterroutinen (SetData, ResData, SetClock, ResClock) bewerkstelligt.

Die Handhabung des Programms ist auch von BASIC aus ohne Schwierigkeiten mög-

lich. Alle zu übertragenden Daten müssen zunächst in einen Ausgabepuffer (PUFFAUS) geschrieben werden. Er faßt maximal 256 Byte und beginnt bei der vorliegenden Version mit $CE00, also dezimal 52 736. In die Speicherzelle ZAHLAUS (hier $CA00 = 51 712) wird die Anzahl der Datenbytes geladen, dagegen übergibt ZAHLEIN (hier $CB00 = 51 968) die Anzahl der erwarteten Antwortbytes. Dann wird die Handling-Routine aufgerufen. Sie beginnt bei $C000. Von BASIC aus muß der Startbefehl also SYS 49 152 lauten.

Nach der Ausführung stehen die empfangenen Daten im Eingabepuffer PUFFEIN zur Verfügung. Er faßt ebenfalls maximal 256 Byte und beginnt hier bei $CF00, also 52 992.

Außerdem wird in der Speicherzelle FEHLNR (hier $CD00 = 52 480) noch der Fehlerstatus übergeben. Bei fehlerfreiem Ablauf der Übertragung steht dort eine 0; ist ein Fehler aufgetreten, dann enthält sie eine Codeziffer. Der Klartext der Fehlermeldung erscheint auf dem Bildschirm, wenn man das Unterprogramm FEHLER anspringt. Es beginnt in unserer Version bei $C11A, also bei dezimal 49 434. Eine mögliche Fehlerquelle ist ein bereits belegter Bus. Dazu gehört die Codeziffer 1. Die Handling-Routine prüft nämlich gemäß den Verkehrsregeln vor jeder Datenübertragung zunächst einmal, ob der I²C-Bus nicht gerade von anderen Teilnehmern besetzt ist. Vor einem Buszugriff müssen sowohl die Taktleitung SCL als auch die Datenleitung mindestens eine ganz bestimmte Zeit lang HIGH-Pegel führen. Sollte ein Buszugriff kurzfristig nicht möglich sein, gibt die Routine mit der Fehlermeldung 1 »Bus bereits belegt« die Kontrolle an das übergeordnete Programmsegment zurück, damit vor einem erneuten Übertragungsversuch zunächst andere Aufgaben wahrgenommen werden können. Der Buszugriff muß danach noch einmal neu eingeleitet werden.

Es empfiehlt sich also, zumindest bei mehreren am I²C-Bus angeschlossenen Bausteinen den Fehlerstatus nach jedem Buszugriff abzufragen.

Eine andere Fehlermöglichkeit ist die Meldung »Acknowledge erwartet« mit der Codeziffer 2. Der Grund dafür könnte sein, daß der Slave-Sender die Datenübertragung durch Weglassen des Quittungsimpulses abgebrochen hat. Es wäre aber auch eine gestörte Übertragung möglich. In diesem Fall sollte man die gleiche Übermittlung noch einmal einleiten und die Ergebnisse vergleichen.

3.4.6 Ein Beispielprogramm zum Experimentieren

Bei allen I²C-Programmen wurden die oben genannten Speicherzellen in den ersten Zeilen des Programms als Variablen festgelegt. Falls die Handling-Routine für einen anderen Speicherbereich neu assembliert wurde, ist so eine leichte Änderung der Benutzersoftware gewährleistet.

Das Programm I2C-EXPER.HILFE ist in BASIC geschrieben und benutzt die erläuterte I²C-Handling-Routine, die natürlich bereits im Speicher stehen muß. Es gibt eine wertvolle Hilfestellung bei den ersten Gehversuchen im I²C-Bussystem.

Nach dem Laden und Starten mit RUN wird man zunächst gefragt, wieviele Daten man über den Bus schicken will. Dann

sind die einzelnen Werte dual anzugeben. Das ist deswegen sinnvoll, weil auch in den einschlägigen Datenblättern meist binäre Angaben vorliegen.

Das Programm fragt anschließend nach der Zahl der erwarteten Antwortbytes. Daraufhin wird automatisch der Eingabepuffer, der später die Antwortbytes enthalten wird, gelöscht und die Busübertragung angestoßen. Als Antwort erhält man eine Liste mit verschiedenen Darstellungsformen der ausgegebenen Bytes. Links in dezimaler, dann in dualer Form und rechts noch als Upper- und Lower-Nibble, also als höher- und niederwertiges Halbbyte zu je 4 Bit. Diese letztere Darstellungsweise wird zum Beispiel beim Auslesen des Uhren-/Kalenderbausteins PCB 8573 benötigt.

Zum Schluß wird noch der Fehlercode mit der entsprechenden Klartextmeldung ausgegeben und gefragt, ob dieselben Daten noch einmal übertragen werden sollen. Dies kann mit der SPACE-Taste angewählt werden; die RUN/STOP-Taste beendet das Programm und jede andere Taste ermöglicht die Eingabe neuer Daten.

3.4.7 Eine I^2C-Echtzeituhr und ein Kalender am C 64

Es folgt ein konkretes Beispiel für den Betrieb des I^2C-Bus am C64. Dazu wird der Uhren-/Kalenderbaustein PCB 8573 vorgestellt und seine Programmierung ausführlich erläutert. Außerdem geht es um die Einbindung der I^2C-Bus-Routine sowohl in BASIC- als auch in Maschinenprogramme.

Im Commodore 64 sind gleich mehrere Uhren untergebracht, die zum Teil von käuflichen Programmen benutzt werden. Die softwaremäßige »Uhr« wird vom Betriebssystem erzeugt und in der Interrupt-Routine betrieben. Diese Eigenart aber ist auch schon ihr großes Manko. Immer, wenn der Interruptbetrieb für kurze Zeit unterbrochen wird, bleibt diese Uhr stehen. Das ist zum Beispiel jedesmal der Fall, wenn ein Zugriff auf die Diskettenstation erfolgt. Es können gravierende Ungenauigkeiten auftreten, die für ernsthafte Anwendungen nicht tragbar sind.

Anders verhält es sich da schon bei den beiden Hardware-Uhren. Sie sitzen jeweils in einem der zwei im C64 enthaltenen CIA-Bausteinen 6526, haben eine Auflösung von einer Zehntelsekunde und werden mit der Netzfrequenz betrieben, die eine hervorragende Langzeitkonstanz aufweist. Allen drei Uhren gemeinsam ist jedoch, daß sie nach dem Einschalten des Systems erst einmal gestellt werden müssen.

Unter den Bausteinen mit I^2C-Schnittstelle befindet sich eine Uhren-/Kalenderschaltung, die mit sehr wenig Strom bei kleiner Spannung auskommt, und die intern auch weiterläuft, wenn der Rechner ausgeschaltet ist. Es handelt sich um den Baustein PCB 8573 (entspricht dem hinsichtlich des Teperaturverhaltens verbesserten PCF 8573), dessen interner Aufbau in Bild 3.9 dargestellt ist. Bild 3.10 zeigt seine Pinbelegung.

Die integrierte Schaltung enthält einen Zeitzähler und ein Alarmregister, jeweils für Minute, Stunde, Tag und Monat. Die Uhr selbst wird von einem Akku versorgt. Aufgrund der verwendeten CMOS-Technologie beträgt die maximale Stromaufnahme nur etwa 10 μA, das heißt, die

Bild 3.9: Innerer Aufbau der Uhren-Kalender-Schaltung

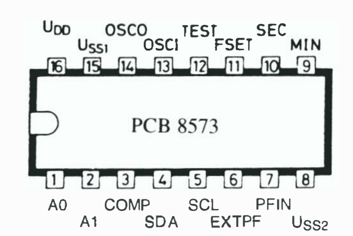


Bild 3.10: Pinbelegung des Schaltkreises

Uhr würde auch noch nach mehr als fünf Jahren laufen, ohne daß der Rechner ein einziges Mal eingeschaltet werden müßte, um den Akku nachzuladen. Die 5-V-Speisespannung wird nur für den Datenverkehr und die Ausgabe von Impulsen benötigt. Dazu dienen auch die Pegelumsetzer PU.

Als Zeitbasis wird ein Quarzoszillator benutzt, dessen Frequenz von 32,768 kHz im Vorteiler durch 2^{15} geteilt wird. Die daraus resultierenden Sekundenimpulse steuern den Zeitzähler an. Dort werden sie von einem – nicht lesbaren – Sekundenzähler summiert und nach jeweils 60 Sekunden an die weiteren Stufen für Minute, Stunde, Tag und Monat übertragen. Wie aus Bild 3.11 zu ersehen, umfaßt der Zählzyklus 24 Stunden. Im Zeitzähler wird automatisch, je nach Monatslänge, ein Zyklus von 28, 30 oder 31 Tagen berücksichtigt.

Die genannten Zähler können über den I^2C-Bus gesetzt und gelesen werden. An den Ausgängen SEC und MIN werden Sekunden- bzw. Minutenimpulse ausgegeben, zum Beispiel für die Steuerung eines Blinkindikators im Sekundentakt. Die Zustände dieser Anschlüsse können auch über den Bus gelesen werden. In das über den I^2C-Bus setz- und lesbare Alarmregister kann eine Schaltzeit gespeichert werden, die mit der aktuellen Zeit ständig

Zähler	Zählerzyklus	Übertrag bei Zählerstand	Stand des Manatszählers
Minuten	00...59	59 → 00	1 ... 12
Stunden	00...23	23 → 00	1 ... 12
Tage	01...28	28 → 01	2
	oder 01...29	29 → 01	2
	01...30	30 → 01	4,6,9,11
	01...31	31 → 01	1,3,5,7,8,10,12
Monate	01...12	12 → 01	

Bild 3.11: Zählzyklen und Überträge des PCB 8573

verglichen wird. Tritt Gleichheit auf, dann setzt der Baustein eine Marke, die als Steuersignal am Bus wie auch am Anschluß COMP zur Verfügung steht, bis sie mit einem speziellen Befehl gelöscht wird. In Abhängigkeit von einer weiteren Marke NODA, die ebenfalls über den Bus beeinflußt werden kann, erfolgt dieser Vergleich wahlweise mit oder ohne Berücksichtigung des Datums.
Die letzte Marke POWF (Powerfail) dient zum Aufdecken von Unzuverlässigkeiten in der Akku-Spannungsversorgung. Hat diese einen bestimmten Grenzwert unterschritten, so wird POWF gesetzt. Erst ein Schreibbefehl in die Zeit- oder Alarmregister löscht die Marke wieder.

3.4.8 Betrieb der Uhren-/Kalenderschaltung am I^2C-Bus

Wie bereits erwähnt, muß mit dem ersten Byte einer Datenübertragung beim I^2C-Bus immer die angesprochene Schaltung

Adresse des PCB 8573
1. Byte nach der Startbedingung:

MSB ... LSB

1	1	0	1	0	A1	A0	R/$\overline{W}$

Bild 3.12: Das Adreßwort des Uhren-Kalender-Bausteins

adressiert werden. Bild 3.12 zeigt das Adreßbyte der Uhren-/Kalenderschaltung PCB 8573. A0 und A1 werden durch zwei Hardwareanschlüsse bestimmt, die auf der vorgestellten Platine gemeinsam auf Masse liegen. Hier muß also in beiden Fällen eine 0 stehen. Zu erkennen ist, daß das niederwertigste Bit des Adreßwortes (R/$\overline{W}$) Einfluß auf die Betriebsart hat. Ist es 0, arbeitet die Schaltung nach der Übertragung des Adreßbyte weiterhin als Empfänger und nimmt das folgende Byte, das sogenannte Subadreßwort, entgegen. Ist R/$\overline{W}$ jedoch 1, schaltet der Baustein beim zweiten Byte auf Senden um und gibt selbst Informationen aus.
Mit dem Subadreßwort kann gezielt ausgewählt werden, welches Register der Schaltung PCB 8573 im folgenden gelesen oder beschrieben werden soll. Mögliche Quellen bzw. Ziele sind etwa Zeitzähler »Minuten«, Alarmregister »Tage« usw.
In der Subadresse können auch Steuerbefehle übertragen werden. Dazu ist das Subadreßwort in 4 Steuerbit (C3...C0) und 4 Adreßbit (B3...B0) aufgeteilt. Bild 3.13 macht das deutlich. Die Subadreßwörter für die einzelnen Funktionen sind in Bild 3.14 zusammengefaßt.

Master-Sender an Uhr/Kalender (Slave-Empfänger)

S	1101 0XX0	A¹	XXXX XXXX	A¹	XXXX XXXX	A¹	– –	XXXX XXXX	A¹	P
	Adreßbyte				1. Datenwort ²)			n.Datenwort ²)		

Subadreßwort

0 C2 C1 C0 0 B2 B1 B0

A¹ = Quittung vom PCF 8573
²) = Autoincrement von B1, B0 durch ACK des Datenworts

Bild 3.13: Ablauf der I²C-Bus-Kommunikation beim PCB 8573

Subadreßwort

Im vorangegangenen Adreßwort muß R/$\overline{W}$ = 0 sein. Das 8-Bit-Subadreßwort besteht aus 4 Steuerbits (C-Bits) und 4 Subadreßbits (B-Bits). x ≙ 1 oder 0

zu Takt	1 C3	2 C2	3 C1	4 C0	5 B3	6 B2	7 B1	8 B0	Tabelle 2 Funktion
	0	0	0	0	0	0	0	0	Lade Adresse für Zeitzähler-Stunde
	0	0	0	0	0	0	0	1	-Minute
	0	0	0	0	0	0	1	0	-Tag
	0	0	0	0	0	0	1	1	-Monat
	0	0	0	0	0	1	0	0	Lade Adresse für Alarmregister-Stunde
	0	0	0	0	0	1	0	1	-Minute
	0	0	0	0	0	1	1	0	-Tag
	0	0	0	0	0	1	1	1	-Monat
	0	0	0	1	X	X	X	X	Lese-Zustand MIN, SEC, NODA, COMP, POWF
	0	0	1	0	X	X	X	X	Verteiler- und Sekundenzähler-Reset (ohne Minutenauf-/-abrundung)
	0	0	1	1	X	X	X	X	Sekundenzähler-Reset ¹) (mit Minutenauf-/-abrundung)
	0	1	0	0	X	X	X	X	Lösche NODA ²)
	0	1	0	1	X	X	X	X	Setze NODA ²)
	0	1	1	0	X	X	X	X	Lösche COMP ²)

¹) Bewirkt eine Zeitkorrektur um maximal ± 30 s. War der Sekundenzähler < 30, so erfährt der Minutenzähler keinen Übertrag. Ein Sekundenzähler > 30 ergibt einen Übertrag in den Minutenzähler (Aufrundung)

²) siehe »Komparator«

Bild 3.14: Bedeutung des Subadreßworts beim PCB 8573

3.4.9 Beispiele für Lesen und Stellen der Uhr

Die folgenden Erläuterungen sollten Sie ruhig einmal mit Hilfe des Programms I2C-EXPER.HILFE nachvollziehen. Nur wenn Sie die Programmierung des Bausteins verstehen, werden Sie selbst alle Möglichkeiten der Schaltung ausnutzen können.

Wenn Sie aus dem Uhren-/Kalenderbaustein Informationen lesen wollen, müssen Sie zunächst genau bestimmen, mit welcher Information zu beginnen ist. Nehmen wir einmal an, uns interessieren die Stunden und Minuten der laufenden Zeit. Dazu müssen wir mittels der Subadresse den Adreßzeiger auf das Stundenregister stellen.

Zuerst einmal wird das Adreßbyte des PCB 8573 für Schreiben gesendet:

11010000 .

Danach folgt das Subadreßbyte für Zeitzähler-Stunde. Dies ermitteln Sie aus Bild 3.14:

00000000

(ganz oben). Die beiden gefundenen Werte werden übermittelt. Der Uhren-/Kalenderbaustein sendet keine Werte zurück, aber nun steht sein interner Adreßzeiger auf dem gewünschten Wert.

Daraufhin fangen wir neu an und senden die Adresse des PCB 8573 zum Lesen:

11010001 ,

also nur 1 Byte. Der Uhren-/Kalenderbaustein antwortet jetzt mit so vielen Datenbytes, wie wir haben wollen. Fordern Sie ruhig einmal zehn an (Im Experimentierprogramm »Anzahl der erwarteten Antwortbytes«: 10).

Das Format, in dem die Ziffern kodiert sind, zeigt Bild 3.15. Es handelt sich hier um BCD-Zahlen. Das ist die Abkürzung für Binary Coded Decimals, also binär codierte Dezimalziffern. In den oberen und unteren 4 Bit des Datenwortes, den beiden Nibbles (Halbbytes), werden dabei jeweils die Zehner und Einer einer zweistelligen Zahl gespeichert. Im ersten Byte erhält man so die ausgelesenen Stunden. Die Dezimalzahl 10 beispielsweise wird nun nicht mehr dargestellt durch die achtstellige Dualzahl

00001010 ,

sondern durch die Darstellung der 1 in der oberen Hälfte des Datenwortes, und einer 0 in der unteren Hälfte, also:

```
1       0
0001    0000
```

zu Takt / Datenwort aus	Zehner				Einer			
	1 UD	2 UC	3 UB	4 UA	5 LD	6 LC	7 LB	8 LA
Stunde	0	0	d	d	d	d	d	d
Minute	0	d	d	d	d	d	d	d
Tag	0	0	d	d	d	d	d	d
Monat	0	0	0	d	d	d	d	d
Zustand von	0	0	0	MIN	SEC	NODA	COMP	POWF

d ≙ Datenbits

Bild 3.15: Codierung der Informationen beim PCB 8573

Im Programm I2C EXPER.HILFE werden unter anderem auch die beiden Halbbytes direkt angezeigt, so daß sich die erhaltene Zahl leicht ablesen läßt.
Sehen Sie sich das nächste empfangene Byte an. Hier tritt bereits eine angenehme Eigenschaft des PCB 8573 zutage: Werden mehrere Bytes ausgelesen, so erhöhen sich die beiden Adreßbytes B0 und B1 (siehe Bild 3.14) selbsttätig durch den Acknowledge-Impuls des Empfängers. So kommt es, daß ohne eine zusätzliche Anweisung der Inhalt des Minutenregisters folgt. Ebenso verhält es sich mit den Tagen und Monaten.

Anschließend wird nicht der Zustand des Adreßbit B2 verändert, sondern es erscheint wieder der Inhalt des Stunden-Registers und so fort, so lange, bis kein Acknowledge mehr den Registerzeiger erhöht und vom Master – in diesem Fall also vom Computer – eine Stoppbedingung folgt.
Die beschriebenen Verhältnisse wurden noch einmal in Bild 3.16 I a) grafisch dargestellt. Wenn Sie zehn Antwortbyte angefordert haben, war die letzte ausgegebene Information diejenige für Minuten. Senden Sie jetzt noch einmal die Bausteinadresse zum Lesen,

`11010001` ,

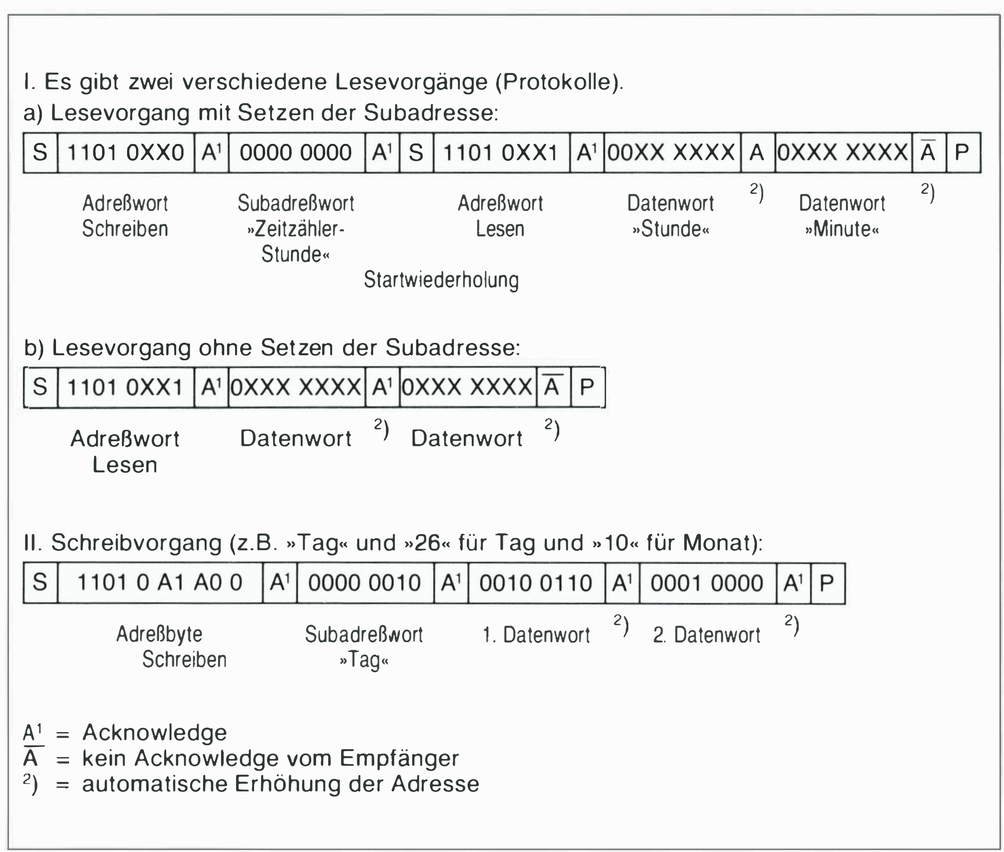


Bild 3.16: Beispiele für Lesen aus dem und Schreiben in den PCB 8573

dann wird genau hier fortgefahren. Um so brauchbare Ergebnisse zu erhalten, muß man also genau die Vorgeschichte des Bausteins kennen. Die zugehörige grafische Darstellung enthält Bild 3.16 I b).
Sehen Sie sich nun Teil II der Abbildung 3.22 an. Es handelt sich um den Schreibvorgang in den Baustein, also das Stellen der Uhr. In unserem Beispiel wollen wir das aktuelle Datum verändern. Dazu müssen wir zunächst wieder den PCB 8573 adressieren. Wir senden also

```
11010000 ,
```

um im nächsten Byte, der Subadresse, den Registerzeiger auf Zeitzähler-Tag zu setzen. Aus Bild 3.14 lesen wir ab

```
00000010
```

(dritte Zeile der Tabelle). Im Gegensatz zum Lesevorgang folgen aber nun weitere Bytes, nämlich die neuen Inhalte der folgenden Register. Auch hier werden die Bits B0 und Bl des Registerzeigers mit jedem Acknowledge-Impuls automatisch um eins erhöht.
In dem abgebildeten Beispiel wird das Datum auf den 26.10. gesetzt. In der linken Hälfte des ersten Datenwortes stehen wieder die Zehner, hier also eine 2 (0010), und in der unteren Hälfte die Einer, also eine 6 (0110). Die Darstellung der 10 ist uns schon von weiter oben bekannt.
Der zweite Teil der Tabelle in Bild 3.14 betrifft die einzelnen Marken. Für ein X kann entweder eine 1 oder eine 0 eingesetzt werden. Wie Bild 3.15 zeigt, sind alle Marken gemeinsam in einem Byte untergebracht. Sie werden ganz analog zu den schon behandelten Registern gelesen, mit der Bytefolge:

```
11010000 : Bausteinadresse zum
           Schreiben
00010000 : Subadresse Lese
           Zustand . . .
```

und anschließend:

```
11010001 : Bausteinadresse'
           zum Lesen.
```

Die Kodierung der Befehle zeigt ebenfalls Bild 3.14. Zum Verändern der unterschiedlichen Marken dienen die übrigen Befehlscodes, die Sie nun selbständig anwenden können.

3.4.10 Die Supportprogramme

Wie die genaue Anwendung der Routine I2C-HANDLING sowie die Handhabung des Uhren-/Kalenderbausteins nun tatsächlich in BASIC- und in Maschinenprogrammen aussieht, das können Sie auch aus den beiden Supportprogrammen sehen.
Das BASIC-Programm I2C-UHR STELLEN macht alle Register des Bausteins leicht zugänglich. Nach dem normalen Laden und Starten mit RUN holt es sich das File I2C-HANDLING.OBJ von der Diskette nach und zeigt zunächst alle augenblicklich eingestellten Werte an. Dazu liest es ständig den Uhren-/Kalenderbaustein über den I^2C-Bus neu aus, so daß tatsächlich die laufende Uhrzeit immer aktuell bleibt. Auch die Sekundenmarke SEC wird ausgelesen und in Form eines blinkenden Doppelpunkts zwischen den Stunden und Minuten der laufenden Uhrzeit dargestellt. Die Unregelmäßigkeit

der Blinkphasen rührt daher, daß der Bildaufbau natürlich nicht genau im Halbsekundenrhythmus erfolgen kann. Aber das zu optimieren, war schließlich nicht die Aufgabe des Programms.

Mit irgendeiner beliebigen Taste – außer Q, die für den Abbruch reserviert ist (Quit) – gelangen Sie zu einem Menü mit den einzelnen veränderbaren Punkten. Bei Anwahl der Punkte 5 und 6 erscheint ein weiteres Menü. Im Unterteil »Statusregister« ist beispielsweise zu wählen, ob der Alarm täglich oder nur zum vorbestimmten Datum erfolgen soll. Hier läßt sich auch die Alarmmarke zurücksetzen.

Sollte Ihnen I2C-UHR STELLEN mit seinen vielen Funktionen zu lang und zu undurchsichtig sein, dann sehen Sie sich bitte das kurze Programm I2C-UHR ANZEIGE an. Hier werden nur einige ausgewählte Standardoperationen benutzt, die ähnlich auch in eigene Programme leicht einzubauen sind. Lassen Sie sich von den vielen Stringoperationen nicht verwirren; die dienen nur einer optisch sauberen Ausgabe der erhaltenen Werte auf dem Bildschirm.

INIT CIA-UHR demonstriert die Einbindung der Routine I2C-HANDLING in ein Maschinenprogramm. Gleichzeitig erfüllt es einen handfesten Zweck: Viele kommerzielle Programme benutzen eine der CIA-Uhren. Wenn diese Uhr aber bei Programmbeginn nicht gestellt wird, wird eine falsche Zeit angezeigt. Die kurze Maschinenroutine INIT CIA-UHR bewirkt, daß sich bei jedem Reset – also auch beim Einschalten des Rechners – die aktuelle Uhrzeit und die vorprogrammierte Alarmzeit in die beiden CIA-Uhren kopiert. Zusätzlich wird der Auswahleingang für 50/60-Hz-Triggerung der Uhren auf die hierzulande gebräuchliche Netzfrequenz von 50 Hz umgestellt, damit die CIA-Uhren auch richtig gehen. Falls die beiden Routinen in einem EPROM abgelegt wurden, das geeignet im Speicherbereich installiert werden muß, wird diese Funktion auch bereits beim Einschalten des Computers ausgeführt. Das Programm verwendet dazu die Modulkennung »CBM80« in den Bytes $8004 bis $8008. Der C 64 fügt daher das Initialisierungsprogramm noch in seine Reset-Routine ein, indem er zu dessen Marke START verzweigt. Dort folgen zunächst noch einige normale Initialisierungsaufrufe, die das System eigentlich erst später ausführen würde. Dann werden die beiden Uhrenschaltungen auf 50 Hz umgestellt, da sie nach einem Reset zunächst immer auf 60 Hz stehen. Gleich danach wird die I^2C-Handling-Routine benutzt. In den Ausgabepuffer kommen die Werte $D0 und 0, um den Zeiger im PCB 8573 auf das Stundenregister zu stellen, und die Routine wird aufgerufen. Die Lösung ist also auch hier ganz analog zu der in einem BASIC-Programm. Da der CIA-Baustein 6526 jedoch nur eine 12-Stunden-Anzeige mit Flag für vormittags und nachmittags besitzt, müssen die Datenformate umgerechnet werden. Die Minuten können direkt übernommen werden.

Anschließend erfolgt der Start der Uhr durch Schreiben in das Zehntelsekunden-Register, und der ganze Vorgang wird noch einmal für das Alarmregister wiederholt. Zur ordnungsgemäßen Funktion muß sich vorher das Programm I2C-HANDLING.OBJ im Speicher befinden.

Das beschriebene Programm AUTO-

INIT.EPROM$8 kann direkt in ein EPROM für die 2* 64-K-EPROM-Karte gebrannt werden. Der Zusatz $8 im Namen deutet an, daß es für den Bereich $8000 gedacht ist, was bei Ausnutzung der Modulkennung Voraussetzung ist. Hinweise für den Einsatz auf der Karte finden Sie in Kapitel 6.4.

Damit man auch während des Programmierens immer genau sehen kann, was die Stunde geschlagen hat, kann jetzt das Programm CIA-UHREINBLENDG verwendet werden, das die Uhrzeit während der Interrupt-Routine immer am oberen rechten Bildschirmrand einblendet.

3.4.11 Weitere Bausteine für den I^2C-Bus

Mit dem Zusatz findet die I^2C-Bus-Familie jetzt problemlos am C64 Anschluß. Interessante Erweiterungen sind ohne großen Aufwand realisierbar. Die Bauteilreihe wird laufend ergänzt und erweitert. Erst kürzlich kam unter der Bezeichnung MAE 8000 von Valvo ein Sprachsynthesizer mit I^2C-Schnittstelle auf den Markt, der in der Markt & Technik-Zeitschrift Design & Elektronik, Ausgabe 19 vom 16.09.1986, Seiten 109ff näher beschrieben wurde. Ein weiterer interessanter Baustein, besonders für Musikfreunde, ist der Sound Fader Control Circuit TEA 6300, mit dem – ohne mechanische Teile – allein über den I^2C-Bus aus drei Stereo-Signalquellen eine ausgewählt und in Klang und Lautstärke beeinflußt werden kann. Durch vier Ausgänge läßt sich programmierbarer Raumklang erzeugen.

Alle I^2C-Bausteine sind unter Ausnutzung der Handling-Routine ähnlich einfach programmierbar, wie die vorgestellte Uhren-/Kalendereinheit.

Kapitel 4
Der serielle Bus

Die am häufigsten benutzte Schnittstelle des C64 ist der serielle Bus. Hier lesen Sie, wie er funktioniert, und was beim Entwurf von Erweiterungsschaltungen zu beachten ist.

4.1 Sparsamkeit als Leitsatz

Leider wurde bei der Konzeption dieses wichtigen Datenkanals, der immerhin auch für die Kommunikation mit der Diskettenstation zuständig ist, tüchtig gespart. Übrig blieb eine abgemagerte Version des parallelen IEC- oder auch IEEE-488-Bus, mit dem die Geräte der CBM-Serie von Commodore ausgerüstet sind. Die Kommunikation mit den angeschlossenen Geräten ähnelt zwar dem parallelen Vorbild, und sogar die Bedienung von Maschinensprache aus geschieht über Routinen, die von der Funktion her identisch mit denen der großen CBMs sind, doch können diese Tatsachen kaum darüber hinwegtrösten, daß beim seriellen Konzept nur eine einzige Datenleitung zur Verfügung steht.
Allgemeine Betrachtungen über Bussysteme finden Sie in Kapitel 3.4.1, und in Kapitel 2.3.4 erfahren Sie, wie sich ohne viel Aufwand eine wesentlich schnellere Parallelübertragung zur Diskettenstation realisieren läßt.

4.2 Die beteiligte Hardware

Der am C64 verfügbare serielle Bus ist ähnlich wie der im vorigen Kapitel beschriebene I^2C-Bus konzipiert. Über Geräteadressen können auch hier mehrere Peripherieeinheiten angesprochen werden. Allerdings ist das System nicht multimasterfähig; es wird von einem einzigen Controller verwaltet, der alle Aktionen koordiniert, nämlich vom C64. Daher ist es auch nicht ohne weiteres möglich, zwei Rechner über den seriellen Bus zu koppeln. Überhaupt dürfen laut Commodore-Spezifikation nur insgesamt fünf Geräte gleichzeitig angeschlossen werden.
Es gibt drei Operationsarten auf dem seriellen Bus: CONTROL (steuern), TALK (senden) und LISTEN (empfangen). Der CONTROLLER koordiniert die Aktionen auf dem Bus, ein TALKER beeinflußt die Datenleitung und ein LISTENER empfängt Informationen.
Der C64 arbeitet sowohl als TALKER, wenn er zum Beispiel Daten zum Drucker überträgt, wie auch als LISTENER, wenn er beispielsweise ein Programm von Diskette lädt. Andere Geräte können entweder LISTENER (Drucker), TALKER (Meßgerät) oder beides sein (Floppy). Auf den

Bus kann immer nur ein TALKER arbeiten, jedoch können durchaus mehrere LISTENER aktiv sein.
Damit Daten zu einem bestimmten Empfänger übertragen werden können, erhält jedes Gerät eine Nummer, also eine Adresse. Beim C64 hat Commodore für den seriellen Bus die Gerätenummern 4 bis 31 reserviert.
Der Anschluß peripherer Geräte geschieht über einen sechspoligen Normstecker, wie er sonst bei Videogeräten oder bei Fernbedienungen von Diaprojektoren gebräuchlich ist. Zumindest an der Floppy VC 1541 befinden sich zwei Anschlußbuchsen, die einfach parallel geschaltet sind, um von dort aus über ein weiteres Kabel das nächste Gerät anschließen zu können. Bei näherer Betrachtung der Steckerbelegung im Anhang C fallen zunächst die beiden Anschlüsse »SER. CLK« und »SER. DATA« auf. Es handelt sich um die vom I^2C-Bus bekannten Signale für Takt und Daten. Auch hier wurde eine Verschaltung mit offenen Kollektoren realisiert. Im Gegensatz zum I^2C-Bus sind jedoch die Spannungspegel auf dem Bus invertiert. Ein LOW-Signal wird durch +5 Volt auf dem Verbindungskabel übertragen, ein HIGH-Impuls entsprechend mit 0 Volt.
Außer der Masseverbindung (GND) sind noch einige zusätzliche Steuerleitungen vorhanden, nämlich die drei Signale »$\overline{\text{RESET}}$:« »SER. SRQ« und »SER. ATN«, deren Funktionen im folgenden noch erläutert werden.
Tabelle 4.1 zeigt, wie die einzelnen Leitungen gesteuert und abgefragt werden können. Der Vollständigkeit halber wurde auch die Floppy VC 1541 berücksichtigt. Dabei ist zu beachten, daß getrennte Anschlüsse für Empfangen und Senden existieren. In die Sendeleitungen sind bei beiden Geräten Inverter mit Open-Collektor-Ausgängen vom Typ 7406 geschaltet. In der Floppy enthalten auch die Empfangsleitungen invertierende Schmitt-Trigger vom Typ 74LS14. Dagegen sind die bezeichneten Eingänge des Computers direkt an die Busleitungen gekoppelt.

4.3 Der Zeitablauf einer Busübertragung

Über den seriellen Bus kann der C64 jeweils ein bestimmtes der angeschlosse-

C64 Adresse		Bit	Bus-Signal	Richtung	VC 1541 Adresse		Bit
$DD00	56576	3	$\overline{\text{ATN}}$	→	$1800	6144	7
$DD00	56576	5	$\overline{\text{DATA}}$	→	$1800	6144	0
$DD00	56576	7		←	$1800	6144	1
$DD00	56576	4	$\overline{\text{CLK}}$	→	$1800	6144	2
$DD00	56576	6		←	$1800	6144	3
$DC0D	56589	4	$\overline{\text{SRQ}}$	←	-	-	-

Tabelle 4.1: Beschaltung des seriellen Bus an C64 und Floppy

nen Geräte auffordern zu empfangen oder zu senden. Das tut er, indem er $\overline{\text{ATN}}$ (Attention = Achtung) auf LOW zieht. Alle angeschlossenen Geräte unterbrechen daraufhin ihre augenblickliche Tätigkeit und gehen in den empfangsbereiten Zustand. Als nächstes legt der CONTROLLER nun Bit für Bit die Geräteadresse auf den Bus – im Fall der Diskettenstation also beispielsweise die 8 – während $\overline{\text{ATN}}$ LOW bleibt. Alle nicht adressierten Geräte ziehen sich danach vom Busbetrieb zurück und setzen ihre unterbrochene Arbeit fort. Das angesprochene Gerät aber muß innerhalb einer Millisekunde reagieren, indem es die Datenleitung nach LOW zieht. Andernfalls nimmt der Computer an, das adressierte Gerät sei nicht am Bus und gibt eine entsprechende Fehlermeldung aus. $\overline{\text{ATN}}$ wird zurückgesetzt und dem Gerät mit TALK bzw. LISTEN mitgeteilt, ob es Daten senden oder empfangen soll.

Ist die Übertragung beendet, so wird das Gerät wieder deadressiert. Dazu sendet der Computer entweder UNTALK, wenn das Gerät vorher Sender war, oder UNLISTEN, sofern es sich im Empfangsmodus befand.

Sollen zum Beispiel Daten von der Floppy geholt werden, muß man nach Geräteadresse und LISTEN den Filenamen und alle wichtigen Fileparameter angeben. Nach UNLISTEN kann der Computer in beliebigen Zeitabständen Daten von der Floppy empfangen. Er legt dazu – wie oben – die Leitung $\overline{\text{ATN}}$ auf LOW, gibt danach die Gerätenummer an und schickt ein TALK über den Bus. Daraufhin stellt die Floppy so lange Daten bereit, bis vom Computer ein UNTALK-Signal kommt. Auch die Floppy kann ihrerseits die Übertragung beenden (zum Beispiel nachdem das letzte Programmbyte übermittelt wurde), indem sie ein EOI (End Of Information) aussendet. Der C64 schließt daraufhin mit einem UNTALK die Übertragung ab.

Da vor jeder Übertragung die Gerätenummer angegeben werden muß, ist es dem CONTROLLER möglich, gleichzeitig mehrere Files auf dem Bus offenzuhalten, um völlig gemischt auf die einzelnen Geräte zugreifen zu können.

Damit die Datenübertragung richtig funktioniert, muß sichergestellt werden, daß ein Folgebyte erst dann übermittelt wird, wenn der Empfänger das vorhergehende bereits verarbeitet hat. Dazu gibt es auch hier Handshake-Signale. Der TALKER teilt durch Freigeben der $\overline{\text{CLK}}$-Leitung mit, daß er Daten senden will. Sobald der LISTENER bereit ist, gibt er seinerseits die $\overline{\text{DATA}}$-Leitung frei, und der TALKER beginnt daraufhin mit der Übertragung eines Byte. Dabei wechselt der Zustand der Datenleitung während der negativen $\overline{\text{CLK}}$-Phase und wird immer beim LOW-HIGH-Übergang des Taktes übernommen. Der LISTENER zieht nach dem vollständigen Empfang eines Byte die $\overline{\text{DATA}}$-Leitung so lange nach LOW, bis er wieder empfangsbereit ist und das nächste Byte folgen kann.

Obwohl sich die Beschreibung dieses Vorgangs kompliziert anhört, kann nur so eine sichere Übertragung mit der geforderten Universalität und der unter den gegebenen Umständen maximal erreichbaren Geschwindigkeit vonstatten gehen. Das Zeitdiagramm (Bild 4.1) macht die Abläufe grafisch deutlich. Tabelle 4.2 enthält die zugehörigen Zeiten.

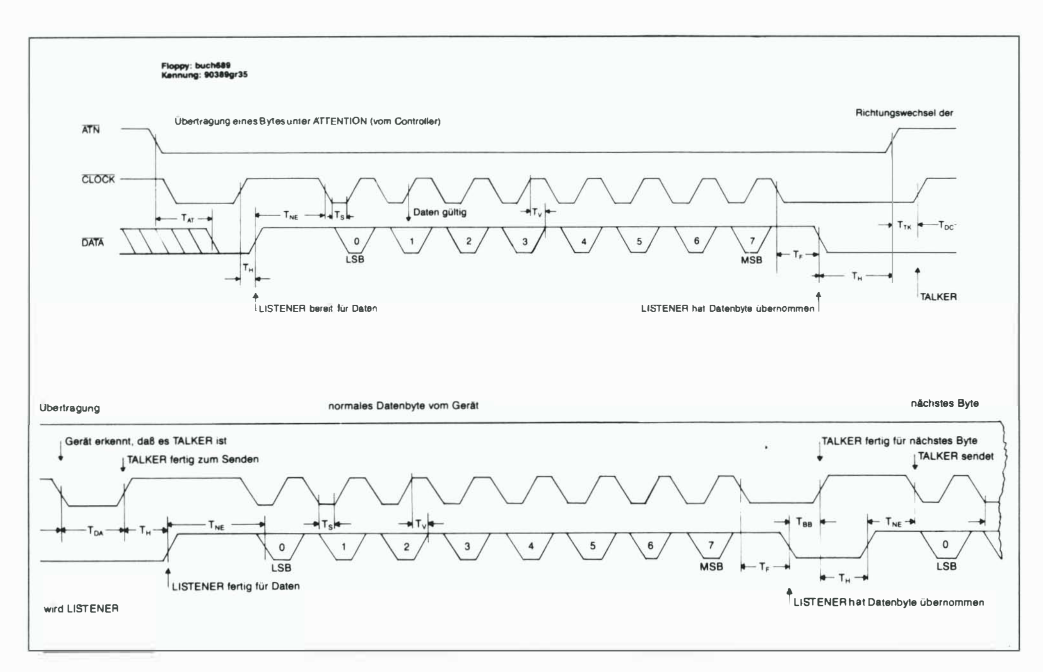


Bild 4.1: Zeitablauf auf dem seriellen Bus

Bezeichnung	Erklärung	Min.	Typ.	Max.
T_{AT}	ATN-Antwortzeit	–	–	1000μs
T_{H}	Listener Haltezeit	0	–	∞
T_{NE}	Maximalzeit bis EOI-Error	–	40μs	200μs
T_{S}	Talker Bit-Einschwingzeit (Set-up)	20μs	70μs	–
T_{V}	Daten gültig (valid)	20μs	–	–
T_{F}	Maximalzeit bis Frame-Error	20μs	–	–
T_{TK}	Talker Haltezeit	0	–	∞
T_{BB}	Zeit zwischen zwei Bytes	–	–	–

Tabelle 4.2: Die Zeiten zu Bild 4.1

Damit wäre die Funktion des ATN-Signals geklärt. Es ist außer am seriellen Bus auch am User-Port (Pin 9) abgreifbar. Die Richtungsangabe in/out am Busstecker bezieht sich auf die Möglichkeit, mit einem ATN-Signal von einem am Bus angeschlossenen Gerät her eine Erweiterung am User-Port anzusprechen. Von den bisher käuflichen Geräten aber ist der C64 das einzige, das die ATN-Leitung beeinflußt. Auch die Floppy VC 1541 kann ATN nur auswerten. Dem C64 selbst ist es nicht möglich, diese Leitung abzufragen.

4.4 Die Reset-Story

Die Funktion der RESET-Leitung dürfte bekannt sein. Es handelt sich um das RESET-Signal des Prozessors, das auch am User-Port (Pin 3) und am Expansion-

Port (Pin C) auftaucht. Mit einem einfachen Taster gegen Masse (zwischen Pin 6 und 2 des seriellen Bus) kann ein Systemreset ausgelöst werden, der dieselbe Funktion erfüllt wie Aus- und Einschalten der Anlage – mit dem wichtigen Unterschied allerdings, daß der größte Teil des Speicherinhalts erhalten bleibt. Eine detaillierte Beschreibung der Vorgänge beim Reset finden Sie in Kapitel 6.4.4. Ein Reset-Taster sollte an keinem C64 fehlen.

Leider gab es bei manchen Exemplaren des C64 Schwierigkeiten mit der $\overline{\text{RESET}}$-Leitung am seriellen Bus, besonders bei mehreren angeschlossenen Geräten. Störimpulse verursachten gelegentlich, daß der Computer einen Reset durchführte. Wie der Querbalken über der Signalbezeichnung andeutet, führt die $\overline{\text{RESET}}$-Leitung normalerweise HIGH-Pegel. Viele Geräte belasten den Anschluß aber so, daß die Spannung bis dicht an die untere Grenze des HIGH-Bereiches absackt. Es muß erreicht werden, daß die $\overline{\text{RESET}}$-Spannung auch bei Belastung nicht zu stark absinkt. Dazu wird ein zusätzlicher Widerstand von etwa 1 Kiloohm zwischen der $\overline{\text{RESET}}$-Leitung und Masse eingefügt. Zur Unterdrückung der Störimpulse kann noch ein Kondensator von 0,1 Mikrofarad parallelgeschaltet werden.

Commodore selbst versuchte das Problem auf andere Weise in den Griff zu bekommen. In neueren Geräten wurde eine Diode in die $\overline{\text{RESET}}$-Leitung eingefügt, so daß die Impulse nur noch in einer Richtung weitergeleitet werden, nämlich vom Computer auf den Bus und nicht entgegengesetzt. Über den seriellen Port läßt sich in diesem Fall kein Computerreset mehr auslösen, auch nicht mit einem Reset-Taster. Als Alternative bleibt dann, den Taster am User-Port anzuschließen, oder ihn gleich fest ins Gehäuse einzubauen.

4.5 Service bitte!

Kommen wir zur anscheinend unwichtigsten Leitung am seriellen Bus, dem Signal $\overline{\text{SRQ}}$. Jedes Gerät am Bus kann theoretisch diesen Anschluß auf LOW ziehen, um dem C64 mitzuteilen, daß es abgefragt werden möchte, erfüllt also in etwa die Funktion eines Handshake-Signals. $\overline{\text{SRQ}}$ wirkt genau wie das RD-Signal des Kassetten-Port auf den $\overline{\text{FLAG}}$-Eingang des CIA # 1 und kann zur Erzeugung eines IRQ dienen. So gut ausgedacht dieses Konzept aber ist, wird es von keinem der üblichen C64-Peripheriegeräte am seriellen Bus benutzt. Selbst die Floppy VC 1541 hat keine Möglichkeit, die $\overline{\text{SRQ}}$-Leitung zu beeinflussen, da diese gar nicht erst an die Elektronik angeschlossen, sondern lediglich zur zweiten Buchse durchgeschleift wurde.

Kapitel 5
Der Audio/Video-Anschluß

Monitorbesitzern ist diese DIN-Buchse bekannt. Aber welche Signale liegen hier überhaupt an und was läßt sich mit ihnen anfangen? Dieses Kapitel vermittelt einige Grundkenntnisse. Zum Schluß geht es um einen externen NF-Verstärker.

5.1 Wie entsteht überhaupt das Bild?

Fernsehgeräte stehen heute in fast allen Haushalten. Die meisten Hersteller von Heimcomputern nutzen diese Tatsache gerne aus und bieten eine Schnittstelle an, die es ermöglicht, den bereits vorhandenen Fernsehempfänger als Ausgabegerät zu benutzen. Auch der C 64 erzeugt sein Bild nach denselben Richtlinien, wie jeder Fernsehsender. Über den eingebauten HF-Modulator ist er in der Lage, am Antenneneingang moderner Fernsehgeräte Anschluß zu finden. Die Fernsehnorm setzt aber auch Standards bei Monitoren, die üblicherweise nach exakt demselben Prinzip arbeiten.

Sehen wir uns kurz an, wie das Bild sowohl beim Fernseher als auch beim üblichen Monitor auf der Mattscheibe entsteht.

Bei der Entwicklung des Fernsehens wollte man bewegte Bilder übertragen. Ähnlich wie beim Film ging man dabei von der Idee aus, dem Betrachter viele einzelne Momentaufnahmen kurz hintereinander zu präsentieren. Im Gegensatz zu auf eine Leinwand projezierten Dias kann aber elektronisch ein gesamtes Bild nicht erzeugt werden. Man setzt es daher aus einzelnen Bildpunkten wie ein Mosaik zusammen. Grob gesagt könnte man für jedes Mosaikteil eine Glühlampe einbauen, die mehr oder weniger hell leuchtet (Bild 5.1). Für jeden Bildpunkt würde in diesem Fall aber mindestens ein Steuerkanal benötigt, was viel zu aufwendig wäre.

Doch schon bei der Auflösung einer fließenden Bewegung in Einzelbilder beim Film läßt sich ja das Auge täuschen. Diese Schwäche wird konsequent ausgenutzt, indem man auch bei jedem Bild die einzelnen Punkte schnell aufeinander folgend zeigt. Dabei wird die Vorlage zeilenweise von links oben nach rechts unten abgetastet (Bild 5.2). Natürlich muß dieser Vorgang sehr schnell ablaufen, damit für den Betrachter ein einheitliches Bild entsteht.

Aus historischen Gründen wurde die Bildfolgefrequenz auf den Wert der Netzfrequenz festgelegt. Daher arbeitet das Fernsehsystem in Europa mit 50 Hz, das in Amerika dagegen mit 60 Hz.

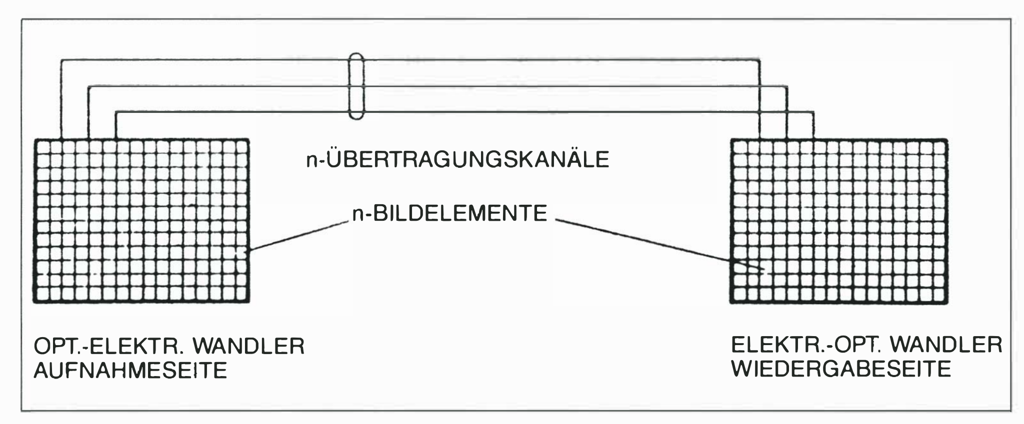


Bild 5.1: Bildübertragung mit Vielfachverbindung

Um bei der Bilderzeugung mit vertretbarem Aufwand auszukommen, erzeugt der Elektronenstrahl unserer Bildröhre einen Lichtpunkt, der durch Magnetfelder auf jeden Punkt der Mattscheibe gesteuert werden kann. Weiterhin ist die Helligkeit dieses Punktes regelbar. Sie ist abhängig von der augenblicklichen Videospannung. Bild 5.3 zeigt die Entstehung dieses Spannungsverlaufs für die gestrichelte Zeile. Hell entspricht einem hohen Spannungswert, dem sogenannten Weißwert, und

Bild 5.2: Fernsehübertragung mit wanderndem Lichtpunkt

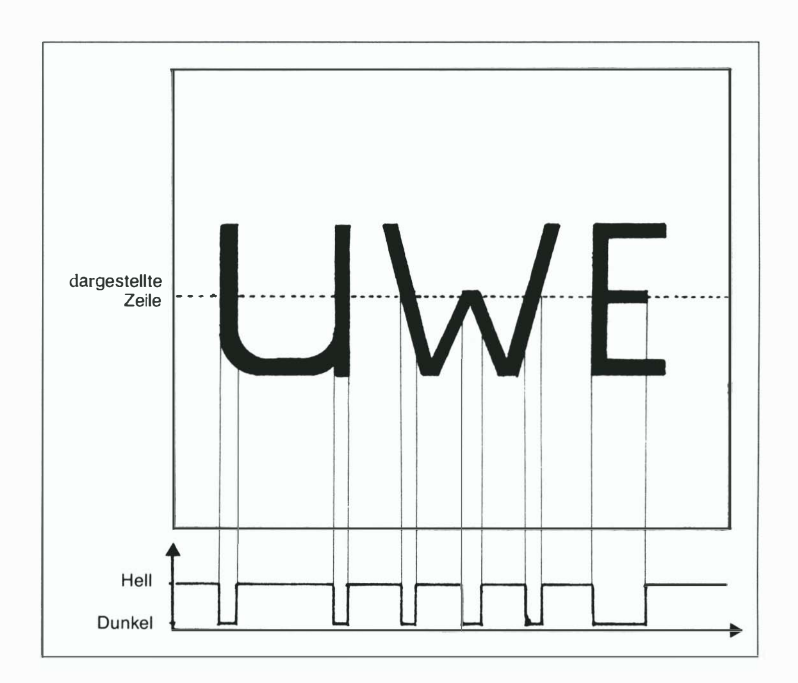


Bild 5.3: Entstehung des Helligkeitssignals

dunkel einem niedrigen. Noch kleinere Spannungen als dieser Schwarzwert haben besondere Synchronisationsimpulse am Anfang jedes Bildes und jeder Zeile. Da durch die Fernsehnorm genau festgelegt ist, in welcher Reihenfolge und Geschwindigkeit die Punkte übertragen werden, kann der Empfänger die Positionierung des Lichtpunktes selbst übernehmen. In Bild 5.2 ist der zusammengesetzte Spannungsverlauf für drei aufeinanderfolgende Zeilen zu erkennen. Ein volles Bild besteht bei uns aus 625 Zeilen.

5.2 Fernseher kontra Monitor

Um das Videosignal senden zu können, wird es in eine hochfrequente Wechselspannung umgewandelt, man sagt auch, es wird einer Trägerfrequenz aufmoduliert. Im Fernsehempfänger geschieht genau das Gegenteil: Das Gemisch wird demoduliert, und man erhält wieder das ursprüngliche Videosignal. Beim Computer verfährt man nicht anders. Der Modulator ist für sich genommen ein kleiner Fernsehsender mit geringer Leistung, der direkt über das Antennenkabel an den Empfänger geschaltet wird.

Am Videosignal geht diese Prozedur allerdings nicht spurlos vorüber. Wer schon einmal mit Hochfrequenz experimentiert hat, weiß, daß einem hier auch die kleinste Ungenauigkeit sehr übel genommen wird. Um scharfe Zeichen zu erhalten, müssen besonders bei Schriftdarstellung abrupte Übergänge zwischen hell und dunkel auftreten (Bild 5.3). In den vielen durchlaufenen Verarbeitungsstufen werden die Übergänge aber nach und nach verwaschen und die Zeichen damit unscharf.

Die Bilddarstellung auf einem ganz nor-

malen Fernsehschirm läßt sich erheblich verbessern, wenn man den Umweg über die Hochfrequenz vermeidet. In letzter Zeit werden dazu verstärkt Fernsehgeräte mit Videoeingang angeboten.

Bei monochromen Monitoren muß der Videoeingang mit dem A/V-Anschluß LUMINANCE verbunden werden. VIDEO OUT enthält auch das Farbsignal und führt zu einem leicht flimmernden Bild.

Farbmonitore können verschiedene Eingänge haben. In den meisten Fällen ist ein FBAS- oder CVBS-Eingang vorhanden. Er wird an den C64-Kontakt VIDEO OUT angeschlossen. Seltener findet man getrennte Eingänge für Helligkeit und Farbe. In diesem Fall liefert LUMINANCE das Helligkeits- und CHROMINANCE das Farbsignal.

Alle Verbindungen sollten über abgeschirmte Kabel erfolgen, wobei das Abschirmgeflecht jeweils an Masse (GND) liegen muß. RGB-Monitore sind nicht direkt anschließbar.

5.3 Externer NF-Verstärker

Was wäre der C64 ohne seinen Sound-Chip? Viele Spielprogramme leben geradezu von der Geräuschkulisse, aber auch in ernsthaften Programmen wird zunehmend mit akustischen Signalen gearbeitet.

Viele Monitore haben den Nachteil, daß sie keinen eingebauten Verstärker besitzen. In solchen Fällen kann leicht über einen einfachen fünfpoligen DIN-Stecker die heimische Stereoanlage an den Pin AUDIO OUT der AV-Buchse angeschlossen werden. Dabei muß die Abschirmung des Kabels an GND liegen. Andernfalls würde starkes Brummen auftreten.

Auf die Dauer ist es aber etwas lästig, immer noch zusätzliche Geräte belegen zu müssen, vor allem, wenn dazu ein Kabel quer durch den Raum führt. Das läßt sich vermeiden, indem ein kleiner Verstärker aufgebaut wird, der die Töne in einem eigenen Lautsprecher hörbar macht. Die Platine für einen solchen Verstärker wurde bereits im Kapitel 2.3.6 im Zusammenhang mit dem Sprachausgabe-Baustein vorgestellt. Der Verstärkerteil kann auch einzeln aufgebaut werden. Hier die Schaltungsbeschreibung:

Auch bei Niederfrequenzverstärkern setzen sich heute immer mehr integrierte Schaltungen durch. So hat man zum Beispiel alle aktiven Bauelemente eines NF-Verstärkers einschließlich der Endstufe in dem IC TBA 800 untergebracht. Dieser Baustein ist sehr populär und findet vor allem in vielen Fernsehgeräten und Kofferradios Verwendung. Er besitzt einen Vorverstärker in Darlington-Schaltung mit hochohmigem Eingangswiderstand, einen Hauptverstärker und eine Endstufe mit zwei npn-Transistoren in Quasikomplementärschaltung mit einer Leistung von 3 Watt. Der Ruhestrom der Endstufe ist intern gegen thermische Einflüsse durch Dioden und Transistoren stabilisiert. Ebenso wird die Mittelspannung der beiden Endtransistoren in der integrierten Schaltung konstant gehalten.

Bild 5.4 zeigt den kompletten NF-Verstärker. Die Außenbeschaltung des IC umfaßt nur wenige Bauelemente und besteht im wesentlichen aus einem Gegenkopplungszweig, der vom Lautsprecherausgang an den Emitter der Eingangs-

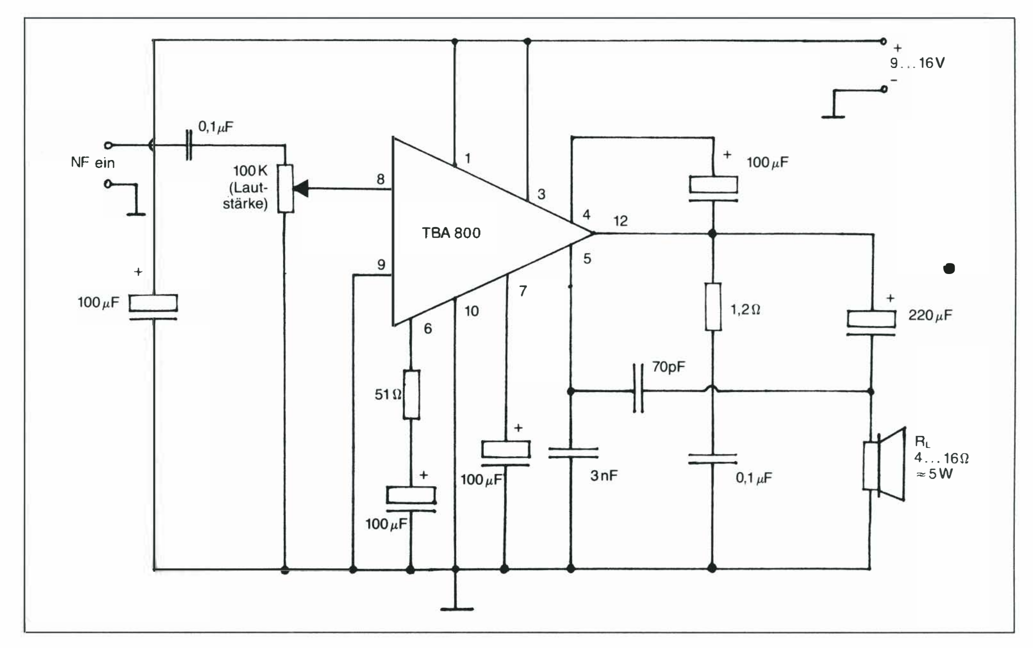


Bild 5.4: Universeller Leistungsverstärker

stufe (Pin 6) geführt wird. Er legt hauptsächlich die Eingangsempfindlichkeit fest. Über das Poti am Eingang kann die Lautstärke reguliert werden.

Das parallel zum Ausgang liegende RC-Glied sorgt in Verbindung mit dem Lautsprecher für einen konstanten Verlauf der Ausgangsimpedanz über den gesamten Niederfrequenzbereich. Mittels des kapazitiven Spannungsteilers am Pin 5 wird in den Vorverstärker eine bei hohen Frequenzen wirksame Gegenkopplung eingespeist. Der Baustein kann in einem weiten Betriebsspannungsbereich von etwa 9 Volt bis 24 Volt eingesetzt werden. Allerdings benötigt er bei voller Ausgangsleistung einen relativ hohen Strom. In jedem Monitor findet sich eine passende Versorgungsspannung und auch etwas Platz zum Einbau der winzigen Platine. Zum Schutz gegen Brummeffekte sollte die Betriebsspannung mit einem Spannungsregler stabilisiert werden. Lesen Sie dazu in Kapitel 7 nach. Notfalls muß ein eigenes Netzteil aufgebaut werden, das die Versorgung übernimmt.

Der NF-Eingang sollte grundsätzlich nur mit abgeschirmtem Kabel beschaltet werden. Dabei liegt das Abschirmgeflecht wieder an Masse.

Kapitel 6
Der Expansion-Port

Wir kommen nun zu dem Erweiterungsstecker mit den meisten Anschlüssen, dem Expansion-Port. Er erlaubt es uns, den C64 gezielt so zu beeinflussen, daß er sich wie ein komplett anderes Gerät verhält. Vorerst sollen aber einige kleine Erweiterungen genügen.

6.1 Anatomie eines Mikrorechners

Die Anschlüsse aller bisher besprochenen Schnittstellen waren vom Programm aus zu steuern. Man konnte jede User-Portleitung einzeln mit einem bestimmten Befehl beispielsweise auf HIGH programmieren und sich ihren Zustand anschließend in Ruhe genau ansehen, die Spannung messen oder auch die Reaktion der angeschlossenen Schaltung beobachten. Anschließend konnte man mit einem weiteren Befehl wieder den ursprünglichen Zustand herstellen. Wir haben Steuerbausteine kennengelernt, wie zum Beispiel die CIAs, mit denen solche Ein-/Ausgabe-Operationen möglich waren, doch wir sind dabei immer am Rand des eigentlichen Computers geblieben.

Wenn wir uns jetzt mit dem Expansion-Port des C64 beschäftigen, ändert sich das grundlegend, denn in den meisten Fällen ist wegen der hohen Arbeitsgeschwindigkeit eine exakte Beobachtung der auf den ersten Blick völlig ungeordnet auftretenden Zustandswechsel ohne teure Meßgeräte völlig unmöglich. Doch glücklicherweise gibt es auch hier – wie überall in der Technik – strenge Gesetzmäßigkeiten und Abläufe, auf die man sich blind verlassen kann.

Jeder Computer besteht aus mehreren Funktionseinheiten, die in festgelegter Weise miteinander verbunden sind. Grundsätzlich enthalten Mikrocomputer neben Einheiten zur Stromversorgung und ähnlichem einen Mikroprozessor, Speicherbausteine und Schaltungen zur Ein- und Ausgabe von Daten. Hier ist jeder Funktionsblock als eigenständiges Gerät zu betrachten.

Der Prozessor (CPU – Central Prozessing Unit = Zentrale Steuereinheit) ist das Gehirn des Computers. Hier laufen alle Befehle ab. Dazu werden Programmanweisungen benötigt, Daten müssen herbeigeholt und Ergebnisse wieder abgespeichert werden. Wegen der hohen Arbeitsgeschwindigkeit herrscht ein reger Datenverkehr zwischen Prozessor und Speicherbausteinen.

Im Speicher sind Unmengen von Daten enthalten. Damit der Prozessor genau angeben kann, welchen Wert er bearbeiten will, erhält jeder Platz, auf dem Daten gespeichert sein können, eine bestimmte Nummer: eine Adresse.

Im Gerät selbst existiert für jede Dualstelle eine Leitung. Beim Prozessor des C64, dem 6510, gibt es 16 Adreßleitungen, so daß insgesamt 65 536 verschiedene Adressen direkt erreichbar sind. Ebenso gibt es acht Datenleitungen (es handelt sich ja um einen 8-Bit-Prozessor), über die jeweils Werte von 0 bis 255 dargestellt werden können. An diese Leitungsbündel sind alle Datenverarbeitungseinheiten des Rechners angeschlossen. Man spricht vom Adreß- bzw. Datenbus des Rechners.

Die Adressen werden immer vom Prozessor erzeugt, seine Adreßanschlüsse sind also Ausgänge. Die Daten dagegen werden mal von einer Speicherzelle geholt und mal zu einem Speicher übertragen. Der Prozessor verfügt daher über eine Steuerleitung, deren logischer Zustand aussagt, ob die CPU gerade Daten liest oder schreibt. Es handelt sich um die Leitung R/$\overline{\text{W}}$ (Read/Write = Schreiben/Lesen).

Erinnern wir uns: Funktionen, die bei LOW-Pegel der entsprechenden Leitung ausgeführt werden, erhalten über ihrer Bezeichnung einen Querbalken. Damit ist die Zuordnung von HIGH und LOW dieses Prozessorausgangs bereits klar. Bei HIGH-Zustand sind die Datenanschlüsse des Prozessors Eingänge, bei LOW dagegen Ausgänge.

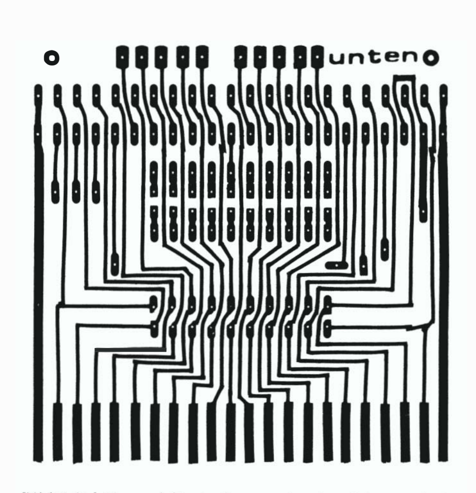
Neben der R/$\overline{\text{W}}$-Leitung gibt es eine Reihe weiterer Steueranschlüsse. Der wichtigste davon führt das Taktsignal $\varnothing_2$. Es wird zur Synchronisierung des Datenaustausches benötigt und gibt dem Computer seine Arbeitsgeschwindigkeit an.

Alle zur Erweiterung des C64 nötigen Anschlüsse sind am Expansion-Port herausgeführt. Dieser Anschluß befindet sich, von hinten gesehen, links auf der Computerrückseite. Anhang C zeigt die Belegung der Kontakte und gibt eine kurze Erläuterung jeder Leitung. An dieser Stelle sei darauf hingewiesen, daß die erste Auflage das C-64-Handbuchs eine falsche Numerierung der Anschlußpins enthielt!

6.2 Der Expansion-Port wird experimentierfreudig

Im Gegensatz zum User- oder Kassetten-Port ragt beim Expansion-Port nicht die Computerplatine zum Aufstecken von Platinendirektsteckern aus dem Gehäuse, sondern hier wurde das Gegenstück gewählt. Um Erweiterungen anschließen zu können, wird eine spezielle Platine benötigt, die alle benutzten Kontakte herstellt. Oft befindet sich auf dieser Platine gleich die gesamte Schaltung. Um jedoch auch frei verdrahtete Experimentierschaltungen anschließen zu können, ist eine Platine nach Bild 6.1 herzustellen.

Es handelt sich um eine doppelseitig kupferkaschierte Ausführung, auf die mit Pfostensteckerleisten direkt Flachbandkabel-Quetschverbinder aufgelötet werden können, über die alle Signale zugänglich sind. Zusätzlich findet im hinteren Teil ein Platinendirektstecker – ein sogenannter Slot – Platz, auf den alle Verbindungen durchgeschleift sind, und über den der Expansion-Port weiterhin für Zusätze benutzt werden kann.



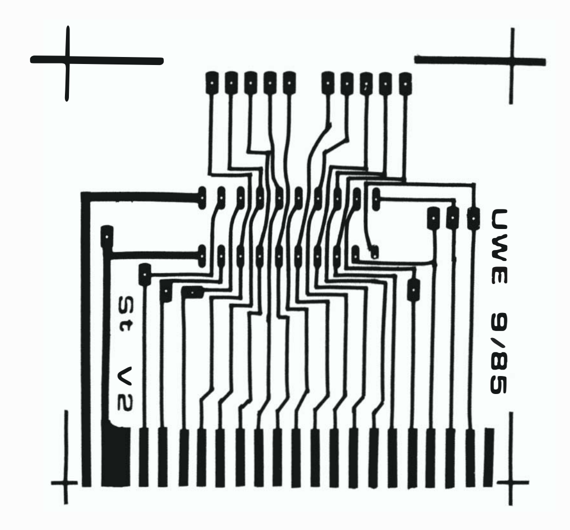


Bild 6.1: Löt- und Bestückungsseite der Adapterplatine

Tabelle 6.1 faßt die Bauteile zusammen. Bei der Bestückung der Platine nach Bild 6.2 sind zunächst die 38 Durchkontaktierungen herzustellen, indem jeweils ein Drähtchen durchgesteckt, von oben und unten verlötet und dicht an der Platine

Expansions-Stecker (Slot)

Flachband-Pfostenstecker 1

Flachband-Pfostenstecker 2

Bild 6.2: Bestückung der Expansions-Steckplatine. Die Pfeile markieren die am Flachbandkabel (Bild 6.3) gekennzeichneten Leiter

2	Pfostensteckerleisten, doppelreihig, 2 x 10 Pole
2	Flachbandkabel-Quetschverbinder, dazu passend
2	Stück Flachbandkabel, 20polig, je max. 20cm
1	Slot, 2 x 22pol
1	doppelseitige Platine nach Bild 6.1

Tabelle 6.1: Bauteile für die Experimentierplatine

abgeschnitten wird. Ansonsten beachten Sie bitte auch hier die Hinweise aus Anhang A, besonders, was die Herstellung doppelseitiger Platinen betrifft.
Die Flachbandkabel dürfen wegen der relativ hohen Arbeitsfrequenz eine Länge von etwa 20 cm nicht überschreiten.
Bild 6.3 zeigt, welche Ader des Kabels dann welches Signal führt. Die Pfeile geben jeweils die Lage im Bestückungsplan (Bild 6.2) an.

6.3 Systemfunktion haarklein

6.3.1 Das Auslesen von ROMs

Für das Verständnis der Vorgänge während eines Speicherzugriffs soll zunächst das Auslesen einer ROM-Speicherzelle erläutert werden. ROMs sind bekanntlich Speicherbausteine mit festem, unveränderbarem Inhalt, aus denen die Daten nur ausgelesen und nicht normal eingeschrie-

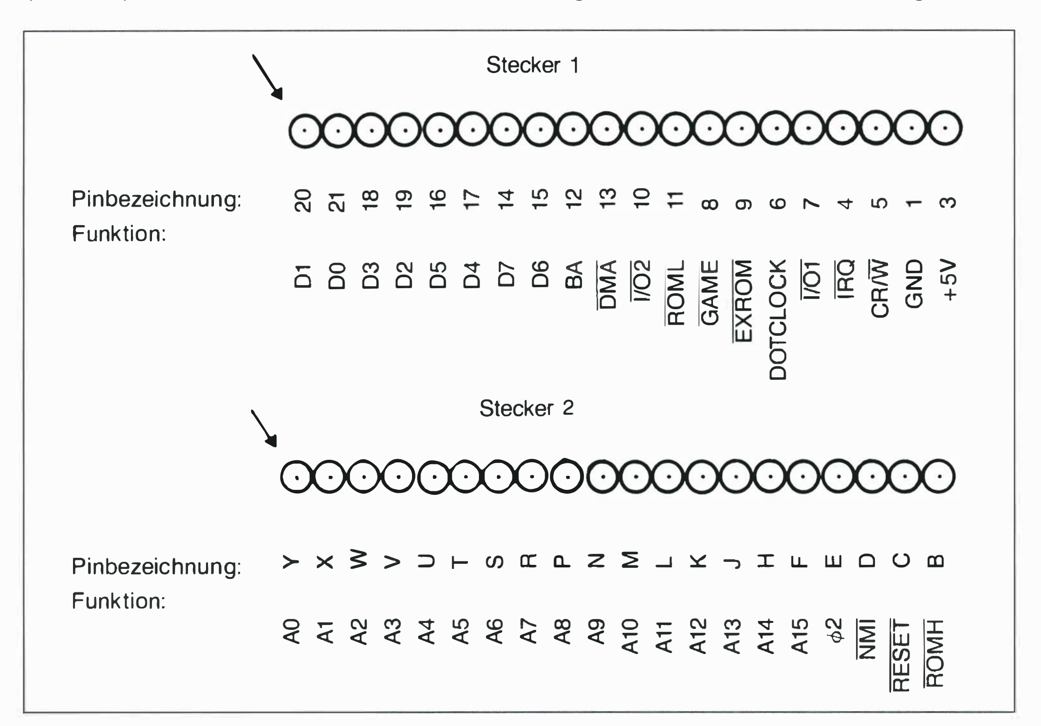


Bild 6.3: Belegung der Flachbandkabel am Expansions-Stecker

ben werden können (ROM: Read Only Memory = Nur-Lesespeicher). Das macht die Betrachtungen besonders einfach. In der Hobbypraxis werden meist EPROMs eingesetzt, die sich durch Bestrahlung mit ultraviolettem Licht wieder löschen lassen. Nach außen hin verhalten sie sich jedoch genau wie fest programmierte ROMs.

Wir betrachten einen solchen Baustein zunächst als schwarzen Kasten, als »Black Box«, in den auf der einen Seite – je nach Typ – 13 bis 16 Adreßleitungen hinein- und auf der anderen Seite 8 Datenleitungen herausführen (Bild 6.4). Zusätzlich gibt es noch zwei Steuereingänge. Die übrigen Anschlüsse für Betriebsspannung, Masse und zum Programmieren sind in diesem Zusammenhang nicht wichtig und können getrost unter den Tisch fallen.

Sehen wir uns nun – stark verlangsamt – die Vorgänge im Computer an, wenn er eine bestimmte Speicherzelle auslesen will.

Zuerst legt der Prozessor auf die Adreßleitungen eine Bitkombination, die der Nummer der gewünschten Zelle entspricht. Für die Speicherzelle 1 ist zum Beispiel nur die erste Adreßleitung HIGH, alle anderen sind LOW. Das EPROM wird noch nicht angesprochen, solange sein Steueranschluß $\overline{CE}$ auf HIGH liegt. $\overline{CE}$ ist die Abkürzung für »Chip Enable« (Bausteinfreigabe) und wird manchmal auch mit $\overline{CS}$ für »Chip Select« (Bausteinauswahl) bezeichnet. Dieser Pin wirkt fast wie ein Ausschalter. Bei HIGH-Pegel befindet sich der Baustein quasi im Winterschlaf. Sogar der Stromverbrauch liegt in dieser »Standby«, »Deselected« oder »Power Down« genannten Betriebsart wesent-

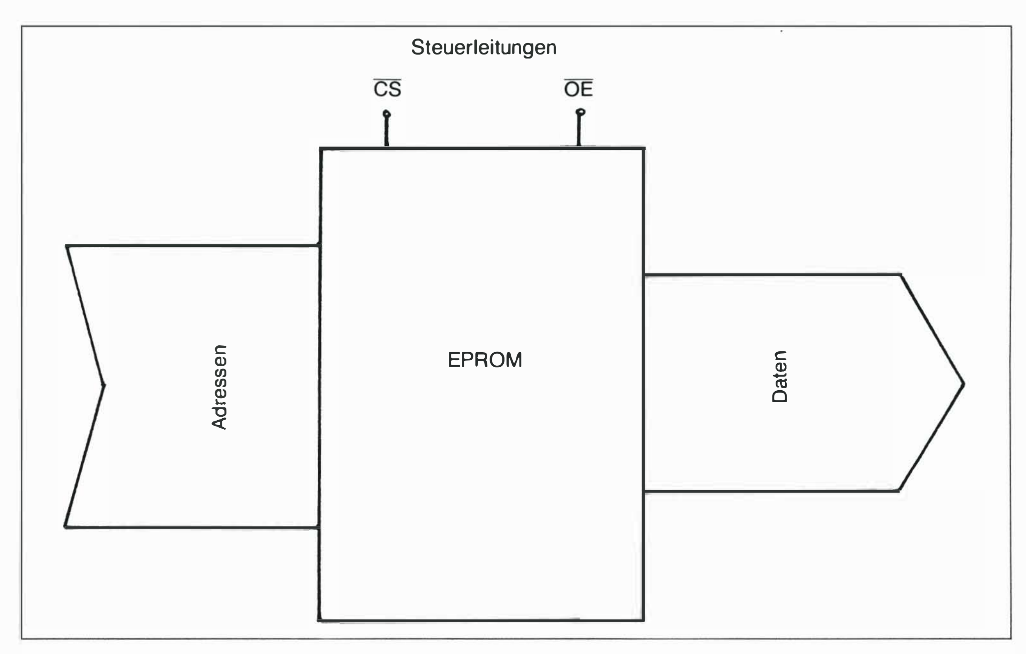


Bild 6.4: EPROM als »Black Box«

lich niedriger, als im eingeschalteten Zustand. Führt $\overline{\mathrm{CE}}$ jedoch LOW-Potential, dann tut sich wenigstens im Inneren des Bausteins etwas: Es wird nachgesehen, welcher Wert unter der angesprochenen Adresse abgelegt ist; das heißt, eigentlich sind es ja mehrere Werte, die 8 Bit für die 8 Datenleitungen nämlich. Hat ein EPROM 8 Kilobyte Speicherkapazität, dann befinden sich in ihm ja 8 * 1024 = 8192 Adressen zu je 8 Bit, also 65 536 einzelne Speicherzellen. Das sind genau 64 Kilobit, denn mit »Kilo« wird in der Computertechnik nicht wie sonst üblich der Faktor 1000 bezeichnet, sondern * 2^{10} = 1024. So kann man sich auch die 64 in der Bezeichnung eines 8-Kilobyte-EPROM 2764 erklären.

Während unserer kleinen Rechnerei hat der Computer überlegt, aus welchem Baustein er die Informationen haben will. Grundsätzlich sind die unteren Adreßleitungen aller Speicherbausteine gleich angeschlossen. Erst die Werte der höherwertigen, nicht an den Speichern liegenden Adreßleitungen entscheiden, welcher Baustein gemeint ist. Die im Computer befindliche Dekodierlogik gibt den ausgewählten Baustein mit LOW am Steueranschluß $\overline{\mathrm{OE}}$ frei. $\overline{\mathrm{OE}}$ ist die Abkürzung für »Output Enable« (Ausgangsfreigabe). Ist dieser Anschluß HIGH, dann haben die acht Datenleitungen des Bausteins einen hochohmigen Zustand, der sich teilnahmslos der jeweiligen Situation auf der Busleitung anpaßt und diese nicht belastet. Es ist gewissermaßen ein dritter Zustand zwischen HIGH und LOW. Man nennt Ausgänge mit dieser Anschaltmöglichkeit daher auch »Tristate-Ausgänge«.

Erst wenn $\overline{\mathrm{OE}}$ LOW wird, schalten die Ausgangstreiber im EPROM durch und geben ihren Zustand an die Anschlußpins weiter. Von der Konstruktion her muß ausgeschlossen sein, daß zwei Bausteine gleichzeitig Informationen auf eine Leitung legen, denn falls die Ausgangsspannungen unterschiedlich sein sollten, ähnelt das einem Tauziehen, und der schwächere der beiden Ausgangstreiber läuft Gefahr, wegen Überlastung zerstört zu werden.

Um die adressierten Daten am Ausgang des EPROM abgreifen zu können, müssen also beide Steueranschlüsse gleichzeitig LOW-Pegel besitzen. Jeder Speicherbaustein benötigt eine gewisse Verzögerungszeit, die vom Anlegen der Adresse bis zum Erscheinen des korrekten Inhalts auf den Datenleitungen vergeht. Es handelt sich um die Zugriffszeit (T_{ACC}: Access Time). Wie erläutert sind EPROMs bei $\overline{\mathrm{CE}}$ = HIGH quasi abgeschaltet. Eine wichtige Zeit ist daher die Chip-Enable-Zeit T_{CE}, die in der Größenordnung der Zugriffszeit liegt. Bild 6.5 macht die Abläufe anschaulich, und Tabelle 6.2 gibt die zugehörigen Zeiten für die wichtigsten EPROMs an. Wie solche Zeitdiagramme gelesen werden, wurde bereits in Kapitel 2.3.6.3 erläutert. Zu erkennen ist, daß die Datenleitungen zunächst noch hochohmig sind. Erst einige Zeit, nachdem $\overline{\mathrm{OE}}$ nach LOW gewechselt hat, senden die Ausgangstreiber ihren Zustand auf den Bus.

Die Zugriffszeiten sind immer als Maximalangaben zu betrachten, das heißt, der Speicherhersteller sichert zu, daß der Baustein die Daten nach der angegebenen Zeit parat hält. Normalerweise wird das schon früher der Fall sein, doch bei der Entwicklung einer Schaltung wird immer vom schlechtesten Fall ausgegangen.

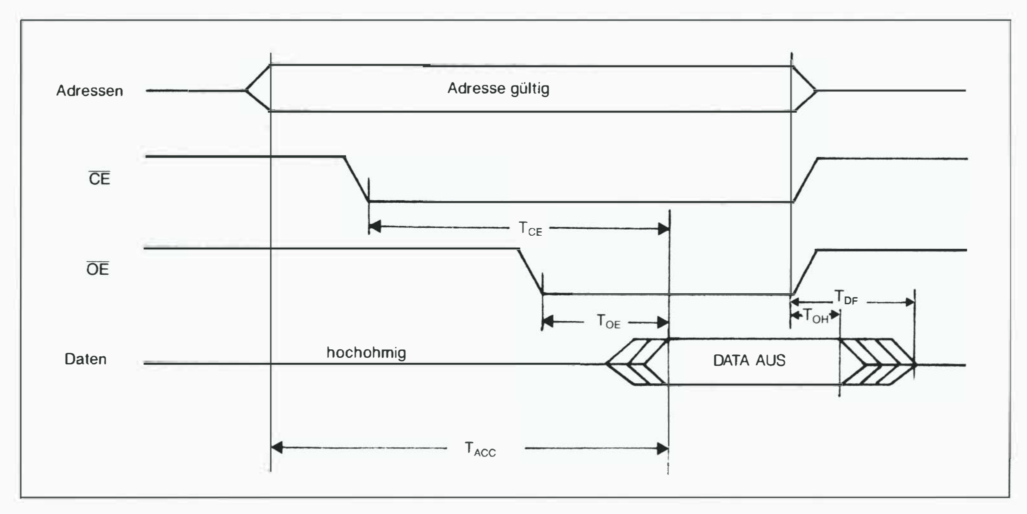


Bild 6.5: Zeitablauf bei der Datenausgabe von EPROMs

EPROM	T_{ACC}	T_{CE}	T_{OE}	T_{OH}	T_{DF}
2716	450	450	120	0	100
2716-1	350	350	120	0	100
2716-2	390	390	120	0	100
2716-5	450	490	160	0	100
2716-6	450	650	200	0	100
P2732 A-2	200	200	70	0	60
P2732 A	250	250	100	0	60
P2732 A-3	300	300	150	0	105
P2732 A-4	450	450	150	0	130
2764 A-1	250	250	100	0	60
2764 A	300	300	120	0	105
2761 A-30	450	450	150	0	130
27128 (-25)	250	250	100	0	85
27128 -30	300	300	120	0	105
27128 -45	450	450	150	0	130
27256	250	250	100	0	60
27256 -3	300	300	120	0	105
27256 -4	450	450	150	0	130
27512 (-25)	250	250	100	0	60
27512 -30	300	300	120	0	105

Alle Angaben in Nanosekunden

Tabelle 6.2: Die Kennwerte der wichtigsten EPROM-Typen nach Intel-Unterlagen

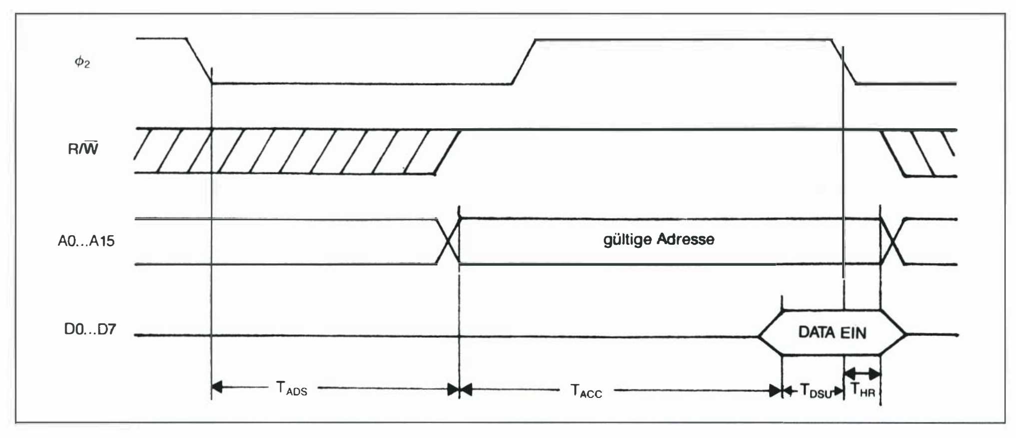


Bild 6.6: Zeitablauf eines Lesezyklus beim 65XX

Außer den Zugriffszeiten sind unter Umständen noch zwei weitere Angaben wichtig. T_{OH} (Output Hold Time = Ausgangshaltezeit) wird von dem Zeitpunkt an gemessen, zu dem eine der Bedingungen $\overline{CE}$ = LOW und $\overline{OE}$ = LOW wegfällt, oder die Adresse wechselt und gibt an, wie lange die Daten noch unverändert anliegen. T_{DF} (Data to Floating Time) bezeichnet die Zeit, bis nach $\overline{OE}$ = HIGH die Datentreiber wieder hochohmig sind, und der Bus von anderen Komponenten belegt werden kann.

6.3.2 Die Prozessor-Speicherzyklen

Nachdem Sie nun wissen, wie ein ROM seine Daten ausgibt, schauen wir uns die Zeitabläufe vom Computer her genauer an. Bild 6.6 zeigt das Ablaufdiagramm, Tabelle 6.3 die zugehörigen Zeiten; vollständigkeitshalber auch die des Prozessors 6502, der in der Floppy VC 1541 arbeitet. Maß aller Dinge ist das Taktsignal $Ø_2$. Im C64 hat es eine Frequenz von nahezu 1 MHz, das heißt eine Periodendauer von gut 1000ns. Während dieser Zeit kann jeweils ein kompletter Schreib- oder Lesezugriff ausgeführt werden. Beim Auslesen

Symbol	6510	6502	Einheit
T_{ADS}	max. 300	max. 300	ns
T_{ACC}	max. 575	max. 500	ns
T_{DSU}	max. 300	min. 100	ns
T_{HW}	min. 10	min. 10	ns
T_{HR}	–	min. 10	ns
T_{MDS}	max. 200	max. 200	ns

Tabelle 6.3: Die Zeiten beim Speicherzugriff nach Bild 6.6 und 6.7

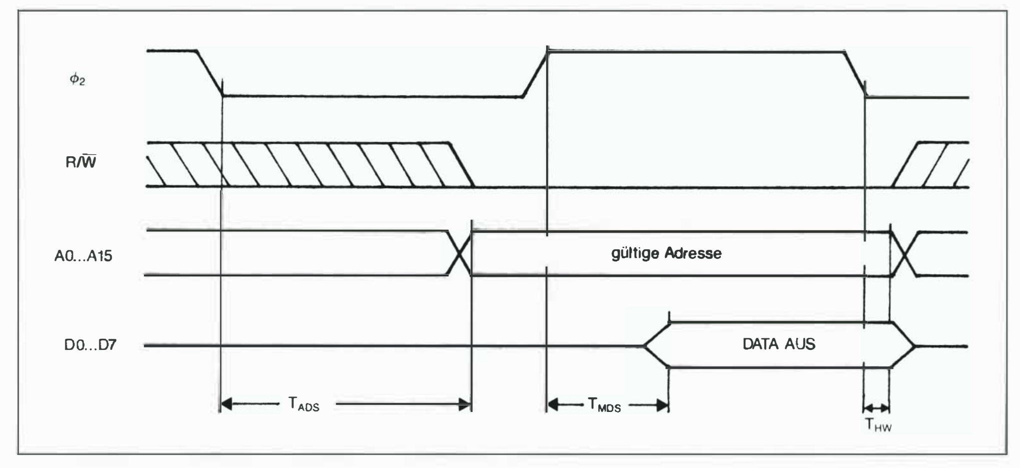


Bild 6.7: Zeitablauf eines Schreibzyklus beim 65XX

einer Speicherzelle geht das folgendermaßen vor sich:

Etwa 300 ns nach der fallenden Flanke von Ø2 legt die CPU die Adresse der gewünschten Speicherzelle auf den Adreßbus. Gleichzeitig zeigt $R/\overline{W}$ durch HIGH-Pegel an, daß es sich um eine Leseoperation handelt. Die Datenanschlüsse des Prozessors werden zu Eingängen und der adressierte Speicher bekommt nun Zeit, den gewünschten Inhalt auf den Datenbus zu legen. Datenübernahmezeitpunkt ist die fallende Flanke von Ø2. In Kapitel 2.3.6.2 wurde bereits genau beschrieben, wie ein D-Flipflop arbeitet. Gleiche Voraussetzungen gelten auch bei der Datenübernahme in den Prozessor. Die Zustände der Datenleitungen dürfen sich während des Intervalls T_{DSU} (Data Setup Time) + T_{HR} (Hold Time bei Read-Operation) nicht ändern. Danach ist der Speicherinhalt übergeben.

Ganz ähnlich läuft die Sache, wenn der Computer Informationen in andere Bausteine schreiben will. Schauen Sie sich in Bild 6.7 den Zeitablauf eines Schreibzyklus an. Wie beim Lesen erscheint die Adresse der gewünschten Speicherzelle erst nach der Data Setup Time (T_{DS}) von maximal 300 ns auf dem Bus. Die Steuerlogik im Computer muß nun dem angesprochenen Baustein durch LOW an $\overline{CE}$ mitteilen, daß er gemeint ist. 200 ns nach der positiven Taktflanke legt der Prozessor die Daten auf den Datenbus und nimmt sie 10 ns nach der negativen Flanke erst wieder fort. In dieser Zeit muß der angesprochene Baustein die Daten übernommen haben. Üblicherweise tut er das an der negativen Flanke des Taktsignals Ø2.

6.3.3 Realisierung von Softwareschaltern

Für die CPU besteht grundsätzlich kein Unterschied zwischen Speichern und Input-/Output-Bausteinen, denn auch die sind letztendlich nichts anderes als Speicherzellen, mit dem Unterschied, daß über ihre Ausgangszustände bestimmte Operationen ausgelöst werden können. Man

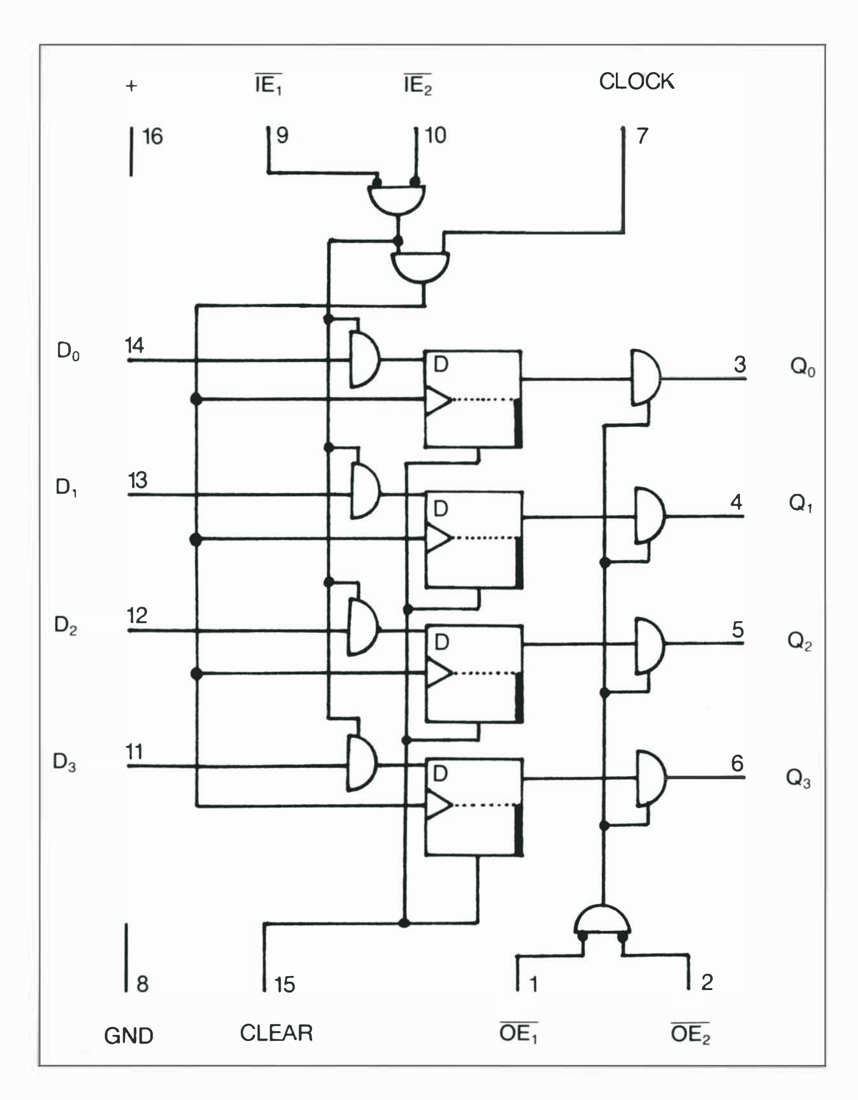


Bild 6.8: Innenschaltung des 4-Bit-D-Registers 74LS173

nennt einzelne Speicherzellen mit bestimmter Bitbreite auch Register.

Als Mini-Beispiel eines Schreib-/Lese-Speichers sollen wieder Flipflops dienen. Bild 6.8 zeigt die Innenschaltung des 74LS173, der vier D-Flipflops und etwas Zusatzlogik enthält. Die Flipflops übernehmen bei jedem LOW-HIGH-Übergang am Anschluß CLOCK den logischen Zustand ihres jeweiligen Eingangs D an den zugehörigen Ausgang Q, falls die gemeinsamen Input-Enable-Eingänge $\overline{IE_1}$ und $\overline{IE_2}$ gleichzeitig LOW sind. An den Ausgängen Q erscheint das Signal aber erst, wenn die beiden Output-Enable-Eingänge $\overline{OE_1}$ und $\overline{OE_2}$ gleichzeitig LOW werden. Andernfalls sind die Treiber – wie bei EPROMs – auch hier wieder hochohmig.

Zusätzlich gibt es noch einen Clear-Eingang, der normalerweise auf LOW liegen soll. Bringt man ihn kurzzeitig auf HIGH, gehen alle Ausgänge sofort auf LOW und bleiben dort.

Die Schaltung stellt in dieser Form bereits einen 4-Bit-Speicher dar. Wir wollen Daten vom Computer in diesen Speicher schreiben lassen. Aus Bild 6.7, dem Zeitdiagramm der Prozessor-Schreiboperation sehen wir, daß die Daten zirka während des letzten Drittels des Zyklus auf dem Bus liegen. Als Übernahmesignal bietet sich die negative Flanke des Taktsignals $Ø_2$ an. Da der 74LS173 jedoch zur Übernahme einen LOW-HIGH-Übergang benötigt, fügen wir einen Inverter ein.

Jetzt müssen wir nur noch dafür sorgen, daß der 4-Bit-Speicher nicht bei jedem $Ø_2$-Impuls neue Daten übernimmt, sondern nur dann, wenn er wirklich gemeint ist. Der C64 macht uns diese Aufgabe besonders leicht, denn am Expansion-Port sind zwei Auswahsignale herausgeführt, die nur dann LOW werden, wenn bestimmte Adressen auf dem Bus liegen. Es handelt sich um den Anschluß $\overline{I/O1}$, der nur dann LOW ist, wenn eine Adresse im Bereich $DE00 bis $DEFF bearbeitet wird, und um $\overline{I/O2}$, für den das gleiche im Bereich $DF00 bis $DFFF gilt. In unserem Fall schalten wir einmal $\overline{I/O2}$ an beide D-Register-Steuereingänge $\overline{IE_1}$ und $\overline{IE_2}$. Damit erreichen wir, daß nur Schreibbefehle auf die Adressen $DE00 bis $DEFF die Flipflops beeinflussen. Natürlich muß dazu CLEAR auf LOW liegen, sonst wären alle vier Flipflop-Ausgänge immer LOW. Schalten wir auch die Ausgangs-Freigabe-Anschlüsse auf LOW, dann werden die eingeschriebenen Zustände gleich an den Ausgängen Q meßbar. Wir haben einen Mini-Portbaustein mit vier Ausgangsleitungen entwickelt!

Da dieser Baustein gleich auf mehrere Adressen reagiert, spricht man von einer unvollständigen Dekodierung. Die unterste Adresse nennt man Basisadresse, alle anderen Bereiche sind Spiegelbereiche. An sich sollten unvollständige Dekodierungen möglichst vermieden werden, da sie bei späteren Erweiterungen Probleme hervorrufen könnten. Um den Bauteilaufwand möglichst gering zu halten, ist diese Schaltungstechnik jedoch besonders bei I/O-Bausteinen üblich. Als Programmierer sollte man immer den Basis-Adreßbereich verwenden, damit die Programme auch bei späteren Systemänderungen noch laufen. Auch die C64-internen Bausteine wie VIC, SID und CIAs tauchen im Adreßbereich mehrmals hintereinander auf. Hier findet sich ein Grund, warum es mit einigen Programmen für den C64 auf dem an sich kompatiblen C128 Probleme gibt. Es werden – aus welchen Gründen auch immer – Spiegelbereiche benutzt, die im C64 genau dieselbe Funktion haben, wie der Basisbereich. Beim C128 wurden jedoch einige zusätzliche Bausteine eingefügt, die nun die ehemaligen Spiegelbereiche anderweitig belegen.

6.4 Der Speicherriese – eine EPROM-Karte für 128 Kilobyte

6.4.1 Eigentlich gar nicht kompliziert

Mit den Kenntnissen aus Kapitel 6.3 gerüstet, sind Sie bereits in der Lage, eine sehr komfortable EPROM-Erweiterung aufzu-

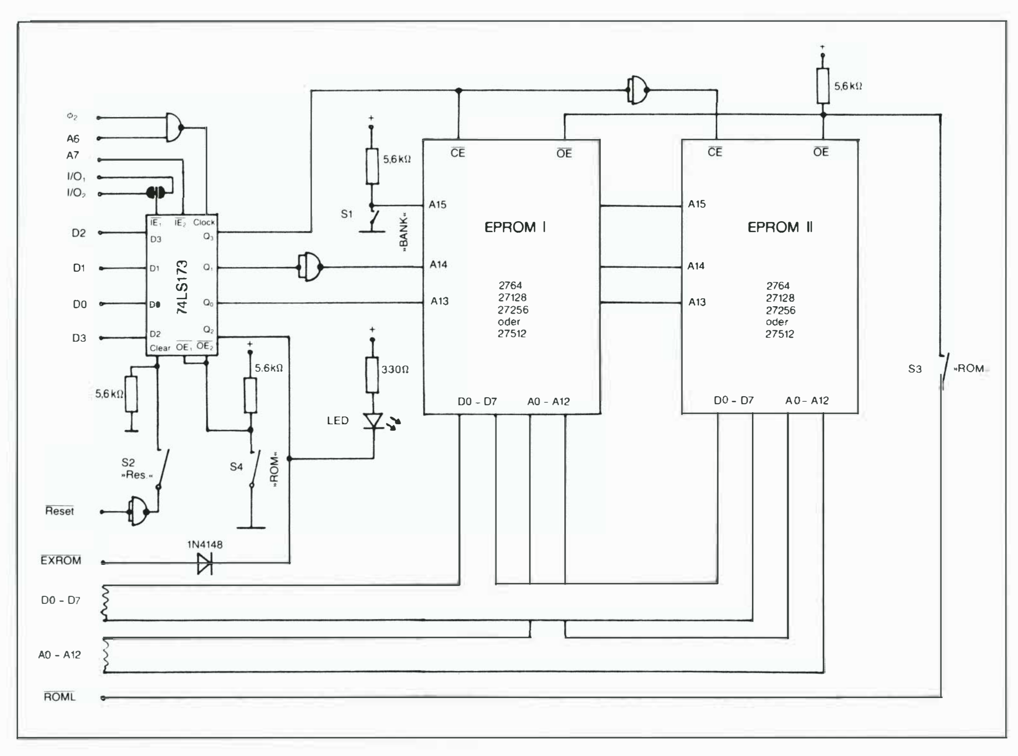


Bild 6.9: Schaltplan 2 x 64-K-EPROM-Karte für C 64

bauen. Wagen wir also einen Blick auf Bild 6.9, den Schaltplan der Zusatzkarte. Ganz links sehen Sie das gerade besprochene Register. Es dient zur Auswahl der verschiedenen EPROM-Bereiche, und seine Ansteuerung wurde im Vergleich zur vorigen Schaltung nur geringfügig erweitert. Fangen wir mit dem CLEAR-Anschluß an. Der C64 benutzt für seinen Reset-Impuls negative Logik, das heißt, ein Reset wird immer dann ausgeführt, wenn die Steuerleitung $\overline{\text{RESET}}$ kurz LOW wird. Um beim Einschalten des Systems genau definierte Werte in den vier Steuerregistern zu erhalten, soll auch der 74LS173 zurückgesetzt werden. Das geschieht durch einfaches Invertieren der Computer-Steuerleitung $\overline{\text{RESET}}$. Für besondere Effekte ist diese Funktion mit dem Dip-Schalter »RES« abschaltbar. Bei dem Widerstand handelt es sich diesmal um eine Pull-down-Schaltung, da über ihn bei offenem Schalter der Anschluß CLEAR auf LOW-Potential festgelegt wird.

Die vier Eingänge des 74LS173 sind mit den Datenleitungen D_0, D_1, D_2 und D_3 beschaltet. Auf der oberen Seite des IC-Symbols wurden im Schaltplan alle Anschlüsse eingetragen, die zur Festlegung des Registers in einen bestimmten Adreßbereich dienen. Um nicht den gesamten $\overline{\text{I/O2}}$-Bereich zu belegen, werden jetzt auch die Adreßleitungen A_6 und A_7 herangezogen. Dadurch sind mehrere Erwei-

terungen in einem einzigen der beiden vordekodierten I/O-Bereiche denkbar. A_7 wurde an $\overline{IE_2}$ gelegt und muß also LOW sein, damit der Baustein reagiert. A_6 dagegen läßt nur bei HIGH-Pegel den Takt $Ø_2$ an den Registereingang CLOCK gelangen.

Aus der Beschaltung können Sie sich die Adressen überlegen, auf die unser Baustein reagieren wird:

Gehen wir einmal von $\overline{I/O2}$ aus. Bei Adressen zwischen $DF00 und $DFFF führt dieser Anschluß immer LOW. Die Umrechnung einer Hexadezimalzahl in die zugehörige Bitkombination ist besonders einfach. Jede Hex-Ziffer entspricht einer Kombination von vier Dualziffern. Daher ist auch die hexadezimale Zahlendarstellung bei maschinennaher Programmierung so beliebt. So erhalten Sie schnell die Adreßleitungszustände im gewünschten Bereich:

```
$D      $F      $0      $0
1101    1111    0000    0000
            bis
$D      $F      $F      $F
1101    1111    1111    1111
```

Zusätzlich müssen noch A_6 HIGH und A_7 LOW sein, also:

```
1101    1111    0100    0000
            bis
1101    1111    0111    1111
```

Das entspricht umgekehrt wieder hexadezimal:

```
$D      $F      $4      $0
$D      $F      $7      $F
```

Damit wirken alle POKE-Befehle in die Adressen 57 152 bis 57 215 auf das Steuerregister. Der Abschnitt wurde bewußt nicht an den Anfang des I/O-Adreßbereichs gelegt, um die Zusammenarbeit mit anderen Erweiterungen möglichst nicht einzuschränken. Die meisten anderen Karten belegen die jeweils ersten Adressen einer der I/O-Seiten, gelegentlich auch die ersten Adressen der zweiten I/O-Hälften, also $DX80 usw. Werden sie angesprochen, würde auch unsere Karte reagieren, wenn ihr Ansprechbereich nicht erst bei $DE40 beginnen würde.

Das Layout der Zusatzplatine enthält einen »Jumper«, mit dem gewählt werden kann, in welchem der beiden Adreßbereiche I/O1 oder I/O2 unser Steuerregister arbeiten soll. Im vorliegenden Layout ist ohne Änderungen $\overline{I/O2}$ beschaltet, da Commodore den Anschluß $\overline{I/O1}$ für die CP/M-Karte vorgesehen hat. Falls die auf der EPROM-Platine vorhandene Brücke durchtrennt und mit etwas Lötzinn eine Verbindung zur anderen Jumperseite hergestellt wird, ist der zuständige Adreßbereich ganz analog $DE40 bis $DE7F, also dezimal 56 896 bis 56 959.

6.4.2 Funktion programmierbar

Per Befehl können die Zustände der vier Ausgänge Q_0 bis Q_3 verändert werden. Jedem einzelnen ist dabei eine bestimmte Aufgabe zugedacht.

Q_2 wirkt auf die $\overline{CE}$-Eingänge der beiden EPROMs. Durch den zwischengeschalteten Inverter wird erreicht, daß immer entweder EPROM I oder EPROM II eingeschaltet ist. Da der C64 hardwarebedingt eigentlich nur externe Erweiterungen von maximal 8 bzw. 16 Kilobyte Größe zuläßt,

stehen lediglich die Adreßleitungen A_0 bis A_{12} zur Verfügung. Daher legen Q_1 und Q_0 gegebenenfalls die Zustände der beiden EPROM-Adreßleitungen A_{14} und A_{13} fest. So können größere EPROMs softwaremäßig in viele 8-Kilobyte-Bänke unterteilt werden, die beliebig umschaltbar einzublenden sind. Will man die volle Speicherkapazität der Karte nutzen, dann muß man 64-Kilobyte-EPROMs mit der Bezeichnung 27 512 einsetzen, deren höchste Adreßleitung A_{15} mit dem Dip-Schalter BANK manuell umschaltbar ist. Diese Adreßleitungen sind gleichzeitig an beide EPROMs angeschlossen. Q_1 – A_{14} enthält dabei einen Inverter, weil dieser Adreßanschluß bei kleineren EPROM-Typen als 27 256 eine andere Funktion hat, und beim Lesen immer HIGH sein muß. Nach einem Reset – also auch nach dem Einschalten – wird Q_1 aber immer LOW sein, so daß eine Umkehrung der logischen Pegel nötig wurde.

Für das Verständnis der Funktion von Q_2 ist eine Erklärung der Hardware nötig:

Der C64 bietet die Möglichkeit, einen Teil seines internen RAM-Speichers durch externe Beschaltung auszublenden, so daß der dadurch freigewordene Adreßraum dann von außen benutzbar ist. Realisiert wird das durch die beiden Expansion-Port-Anschlüsse $\overline{\text{EXROM}}$ und $\overline{\text{GAME}}$. Sie sind in unbeschaltetem Zustand im C64 durch Pull-Up-Widerstände auf HIGH festgelegt. Wird $\overline{\text{EXROM}}$ von außen auf LOW gelegt, dann schaltet der Computer sein internes RAM im Bereich \$8000 bis \$9FFF (32 768 bis 40 959) ab und kennzeichnet Zugriffe auf diesen Bereich, indem er gegebenenfalls den Expansion-Port-Ausgang $\overline{\text{ROML}}$ auf LOW setzt.

Ebenso schaltet LOW an $\overline{\text{GAME}}$ den Bereich \$A000 bis \$BFFF (40 960 bis 49 151) ab und meldet mit LOW an $\overline{\text{ROMH}}$, wenn eine dieser Adressen auf dem Bus liegt. $\overline{\text{ROML}}$ und $\overline{\text{ROMH}}$ bleiben aber durchgehend HIGH, wenn auch ihr zugehöriger Anschluß $\overline{\text{EXROM}}$ bzw. $\overline{\text{GAME}}$ HIGH ist. Tabelle 6.4 zeigt die vier Beschaltungsmöglichkeiten von $\overline{\text{EXROM}}$ und $\overline{\text{GAME}}$ und ihre Wirkung.

Diese Vorkehrungen geben uns die Möglichkeit, ohne viel Hardwareaufwand und ohne Eingriffe in den Rechner, einen Teil des Adreßraums von außen zu beschalten.

Da in unserer EPROM-Karte auch BASIC-Programme laufen sollten, belegt sie nur den unteren 8-Kilobyte-Bereich von \$8000 bis \$9FFF. Dazu muß der am Expansion-Port befindliche Anschluß $\overline{\text{EXROM}}$ von der Erweiterung auf LOW-Pegel gezogen werden. Wie Sie sehen, geschieht das, wenn in das unterste Flipflop D_2/Q_2 eine 0 geschrieben wird. Gleichzeitig zeigt die

$\overline{\text{EXROM}}$	$\overline{\text{GAME}}$	ausgeblendeter Bereich	$\overline{\text{CE}}$-Signal an
1	1	keiner (Normalzustand)	kein Signal
1	0	v e r b o t e n	
0	1	\$8000 – \$9FFF	$\overline{\text{ROML}}$
0	0	\$8000 – \$BFFF	$\overline{\text{ROML}}$/$\overline{\text{ROMH}}$

Tabelle 6.4: Die Beschaltungsmöglichkeiten von $\overline{\text{EXROM}}$ und $\overline{\text{GAME}}$

Leuchtdiode an, daß augenblicklich nicht RAM, sondern EPROM im fraglichen Adreßbereich aktiv ist. Die eingefügte Universaldiode gewährleistet, daß beim Einsatz von mehreren Erweiterungen auch wirklich nur von der jeweils aktiven Karte ein Leuchtsignal kommt. Sollte $\overline{\text{EXROM}}$ von einer anderen Einheit nach LOW gezogen werden, dann verhindert die Universaldiode einen Stromfluß durch die LED.

Falls $\overline{\text{EXROM}}$ LOW ist, wird das Auswahlsignal für den Adreßbereich $8000 bis $9FFF am Ausgang $\overline{\text{ROML}}$ zugänglich gemacht; das heißt, immer wenn auf dem Adreßbus eine der zugehörigen Bitkombinationen erscheint, wird die genannte Leitung LOW. Damit kann dieses Signal direkt zur Freigabe der EPROM-Ausgänge benutzt werden. Zur Erinnerung: Es gibt immer das EPROM Daten aus, dessen $\overline{\text{CE}}$-Eingang ebenfalls LOW ist.

Um die Erweiterung ohne Abziehen vom Expansion-Port ganz abschaltbar zu machen, wurde auch $\overline{\text{ROML}}$ mittels des Schalters S_3 unterbrechbar gemacht. Über den Pull-Up-Widerstand liegen die $\overline{\text{OE}}$-Eingänge der beiden EPROMs dann in jedem Fall auf HIGH. Ihre Ausgänge sind also im Normalfall hochohmig.

Zusammenfassend kann folgende Zuordnung beim Einschreiben eines Steuerwortes angegeben werden:

D_0 und D_1 besorgen die Auswahl der EPROM-Bank beim Einsatz von größeren EPROMs als 2764. Ein 27 128 enthält zwei Bänke zu je 8 Kilobyte, ein 27 256 vier und ein 27 512 sogar acht.

D_2 bestimmt, welches EPROM eingeschaltet ist. 0: EPROM I, 1: EPROM II.

D_3 schließlich entscheidet, ob die Karte überhaupt aktiv ist. 0 schaltet sie ein, 1 dagegen aus.

6.4.3 Die softwaremäßige Auswahl der EPROM-Bänke

Tabelle 6.5 zeigt noch einmal grafisch alle 16 möglichen Kombinationen der vier signifikanten Registerbits zusammen mit ihrem dezimalen Wert und ihrer Funktion. Es wird deutlich, daß in der unteren Hälfte der Tabelle, in der D_3 immer 1 ist, die EPROM-Karte durchgehend ausgeschaltet bleibt, unabhängig davon, welchen Wert die anderen Register haben. In der oberen Hälfte ist entweder EPROM I oder EPROM II angesprochen, je nach Zustand von D_2. Die beiden niederwertigsten Bits D_0 und D_1 legen schließlich eine der jeweils maximal vier möglichen EPROM-Bänke fest.

Ist die Karte – wie im Layout voreingestellt – auf $\overline{\text{I/O2}}$ gejumpert, dann ist die Basisadresse des Schaltregisters 57 152. Mit dem Befehl

```
POKE 57152,0
```

setzt man alle 4 Bit des Steuerregisters auf 0, denjenigen Wert, der in der ersten Zeile von Tabelle 6.5 dargestellt ist. So wird also bereits Bank 0 in EPROM I eingeschaltet. Auf ähnliche Weise kann man beispielsweise Bank 2 in EPROM II erreichen. Aus Tabelle 6.5 erhalten Sie die siebte Zeile

```
POKE 57152,6.
```

Ausschalten ist zum Beispiel mit

```
POKE 57152,8
```

möglich. Wichtig ist nur, daß Bit 3 gesetzt

Wert	D0	D1	D2	D3	Bank	EPROM
0	0	0	0	0	0	I
1	I	0	0	0	1	
2	0	I	0	0	2	
3	I	I	0	0	3	
4	0	0	I	0	0	II
5	I	0	I	0	1	
6	0	I	I	0	2	
7	I	I	I	0	3	
8	0	0	0	I	0	ROMs ausgeschaltet RAM eingeschaltet
9	I	0	0	I	1	
10	0	I	0	I	2	
11	I	I	0	I	3	
12	0	0	I	I	0	
13	I	0	I	I	1	
14	0	I	I	I	2	
15	I	I	I	I	3	

Tabelle 6.5: Die Zuordnung der Auswahlbefehle zu den EPROM-Bänken

wird. Die gleiche Aufgabe erfüllt ebenfalls das Setzen aller Bits mit

```
POKE 57152,255.
```

Selbstverständlich kann man die oberen Bänke nur in genügend großen EPROMs erreichen. Tabelle 6.6 gibt einen Überblick über die einsetzbaren EPROM-Typen und

Bank		Bereich 0	Bereich 1	Bereich 2	Bereich 3
Datenbit	D0	0	1	0	1
Datenbit	D1	0	0	1	1
2764		0000 - 1FFF			
27128		0000 - 1FFF	2000 - 3FFF		
27256		4000 - 5FFF	6000 - 7FFF	0000 - 1FFF	2000 - 3FFF
27512	Bank 1	4000 - 5FFF	6000 - 7FFF	0000 - 1FFF	2000 - 3FFF
27512	Bank 2	C000 - DFFF	E000 - FFFF	8000 - 9FFF	A000 -BFFF

Tabelle 6.6: Adreßbereiche der EPROM-Bänke

die genaue adreßmäßige Lage der einzelnen jeweils ansprechbaren Bereiche. Besonders ist zu beachten, daß beim 27 256 und 27 512 die Bereiche 2 und 3 vertauscht sind. Das kommt durch die Invertierung von A_{14} aus den schon weiter oben genannten Gründen. Beim 64-Kilobyte-EPROM 27 512 kann mit dem Dip-Schalter S_1 zwischen oberer und unterer Hälfte umgeschaltet werden. Bei allen anderen EPROM-Typen sollte dieser Schalter geöffnet bleiben. Ansonsten ist keine Änderung oder Voreinstellung beim Wechsel von EPROM-Typen zu beachten.

6.4.4 Der Einsatz der EPROM-Karte im täglichen Gebrauch

Nach dem Einschalten des Rechners befindet sich zunächst immer EPROM I, Bank 0 im Adreßbereich $8000 bis $9FFF. Hier kann ein Menüprogramm untergebracht werden, das per Modulkennung gleich automatisch gestartet wird. Dazu muß bekannt sein, welches Verhalten der C 64 einnimmt. Beim Hochfahren der Betriebsspannung sorgt eine Verzögerungsschaltung dafür, daß die Leitung $\overline{\text{RESET}}$ erst nach HIGH geht, wenn bereits alle anderen Leitungen ihren Anfangszustand eingenommen haben. Dadurch werden alle Register in den Steuerbausteinen in eine definierte Lage gebracht, und auch der Prozessor kann planmäßig seine Programmabarbeitung beginnen. Dazu holt er sich die Startadresse in den Speicherzellen $FFFC und $FFFD ab. Es ist der Reset-Vektor, der beim C64 $FCE2 enthält. Dort beginnt eine Routine, die zunächst die maskierbaren Interrupts sperrt, den Stack initialisiert, das Dezimalflag löscht und dann in ein Unterprogramm (ab $FD02) verzweigt, das testet, ob in den Speicherzellen $8004 bis $8008 der Inhalt »CBM80« steht. Tabelle 6.7 gibt die genaue Kodierung an.

Falls der C 64 diese Zeichen findet, springt er zu der Adresse, die in den beiden Stellen $8000 (Lowbyte) und $8001 (Highbyte) angegeben ist; falls nicht, initialisiert er den Rechner für BASIC und führt einen BASIC-Warmstart durch. Normalerweise ist RAM im betrachteten Speicherbereich selektiert. Beim Einschalten der Betriebsspannung ergeben sich zufällige Werte in den einzelnen Zellen, und es wäre schon ein großer Zufall, wenn die fünf Adressen ausgerechnet die angegebenen fünf Zeichen in der richtigen Reihenfolge enthielten.

Also existieren nur zwei realistische Möglichkeiten, wie die Kennung zustande

Adresse		Inhalt		
hexadezimal	dezimal	hexadezimal	dezimal	Zeichen
$8004	32772	$C3	195	C
$8005	32773	$C2	194	B
$8006	32774	$CD	205	M
$8007	32775	$38	56	8
$8008	32776	$30	48	0

Tabelle 6.7: Die Kodierung der Modulstart-Kennung

kommen konnte. Die erste ist, daß ein Programm vorher irgendwie ins RAM geschrieben wurde und jetzt mit einem Reset gestartet wird, ohne den Rechner zwischendurch auszuschalten. Die zweite setzt voraus, daß extern ein ROM-Modul angesteckt wurde, das im fraglichen Bereich das RAM ersetzt. In beiden Fällen kann durch Betätigen der Reset-Taste ein automatischer Programmstart bewirkt werden.

Unsere EPROM-Karte könnte im EPROM I, Bank 0 ein Menü enthalten, aus dem man per Tastendruck entweder zu einem in einer beliebigen 8-Kilobyte-Bank gespeicherten Programm oder zur Original-Einschaltmeldung und Speicheraufteilung des C64 verzweigen kann. Falls ein Programm gewählt wurde, schaltet eine kleine Routine die entsprechende Bank ein, kopiert die Daten an den BASIC-Anfang, schaltet die EPROM-Karte wieder aus und startet das Programm. Zum Beispiel kann so ein Schnelladeprogramm für Disk-Betrieb eingefügt werden.

Im Zusammenhang mit der I^2C-Uhr wurde bereits in Kapitel 3.4.10 ein entsprechendes Programm mit Modulkennung vorgestellt. Beim Brennen des File AUTO-INIT.EPROM$8 (beispielsweise in ein 2764) und Einsetzen als EPROM I führt der C64 nach dem Einschalten und bei jedem Reset einen Modulstart aus, wobei er die aktuelle Zeit in beide CIA-Uhren übernimmt und anschließend wieder auf RAM umblendet.

Für Anwender, die sich das individuelle Erstellen von Autostartprogrammen und die damit verbundene Programmierarbeit nicht zutrauen, gibt es »Modulgeneratoren«. Die Diskette enthält einen speziell für diese Karte geschriebenen Generator, der allerdings zwei Laufwerke voraussetzt. Universell ist der MODUL MASTER von Softeis, der komfortabel die Umarbeitung von BASIC- und Maschinenprogrammen in EPROM-Dateien vornimmt, und die Organisation inklusive Menü selbständig erledigt. Er enthält einen Editor zum Erstellen des Einschaltbildes. (Bezugsadresse: Computertechnik Zeno Zaporowski, Vinckestraße 4, 5800 Hagen 1)

Selbstverständlich ist es auch möglich, fertige Autostart-EPROMs in einem anderen Bereich als Bank 0 auf der Karte unterzubringen. Die jeweilige Bank wird mit

```
POKE 57152,...
```

eingeschaltet. Danach erfolgt der Start durch einen Software-Reset mittels

```
SYS 64738 .
```

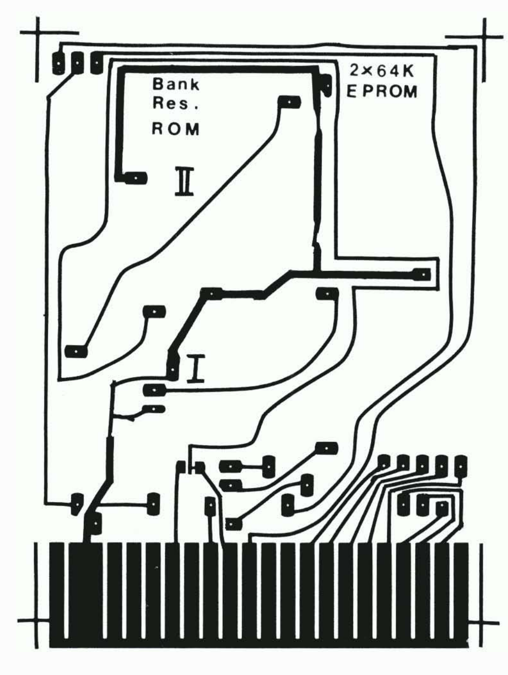
In diesem Fall wird nicht auf EPROM I, Bank 0 zurückgeschaltet, da ja die Reset-Leitung nicht LOW wurde. Der Computer verhält sich so, als ob nur dieses eine 8-Kilobyte-EPROM vorhanden wäre. Diese Vorgänge können auch vom Menüprogramm erledigt werden.

Eine Möglichkeit, das einmal mit dem POKE-Befehl eingeschaltete EPROM auch beizubehalten, wenn die Reset-Taste gedrückt wird, bietet sich durch Öffnen des Schalters »Res.« (S_2).

Das eben erwähnte File AUTOINIT.-EPROM$8 enthält zur Demonstration dieser Eigenschaften ab Adresse $9000 noch einen Monitor, der durch Einschalten von Bank 0 per Hand eingeblendet wird und dann mittels

```
SYS 9*4096
```

zu starten ist.



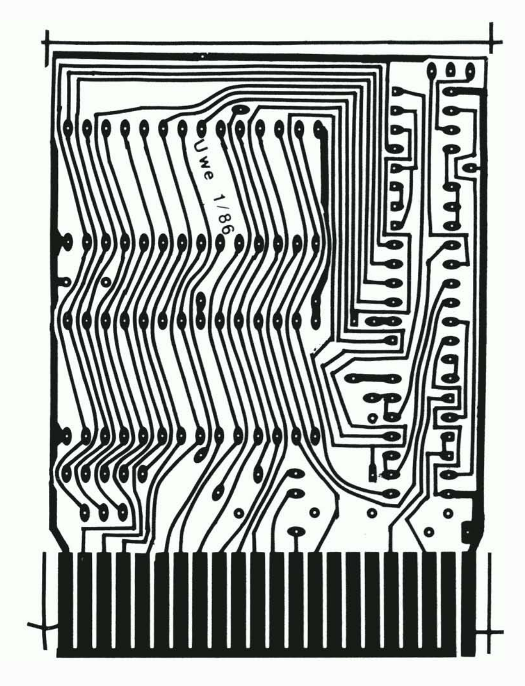


Bild 6.10: Layout für die EPROM-Karte

Soll die EPROM-Karte ganz abgeschaltet werden, dann sind beide Dip-Schalter S_3 und S_4 zu öffnen.

6.4.5 Der Aufbau der EPROM-Karte

Bild 6.10 zeigt Löt- und Bestückungsseite der 2*64-Kbyte EPROM-Karte. Bei der Herstellung und beim Bestücken der Platine beachten Sie bitte den Bestückungsplan (Bild 6.11) und die Hinweise im Anhang A. Die angezeichneten Maße müssen zumindest an der Steckseite genau eingehalten werden. Lieber läßt man sie zunächst etwas reichlich und feilt dann an beiden Seiten so viel ab, bis die Karte exakt in den Expansion-Stecker paßt. Bei zu geringer Breite könnte die Platine verrutschen und zwischen den einzelnen Kontakten Kurzschlüsse hervorrufen.

Die unvermeidlichen Durchkontaktierungen sind im Bestückungsplan durch dicke Punkte gekennzeichnet und sollten vor allen anderen Teilen eingelötet werden. Beachten Sie dabei jedoch, daß gelegentlich auch die Anschlußdrähtchen einiger diskreter Bauelemente durch beidseitiges Verlöten zur Durchkontaktierung herangezogen werden. Als Dip-Schalter sollte eine möglichst kleine Ausführung gewählt werden. Achten Sie besonders auf die richtige Polung der Diode. Es ist empfehlenswert, sämtliche ICs – auch die beiden TTL-Bausteine – zu sockeln.

Falls die Zuordnung der Platine nach $\overline{I/O1}$ geändert werden soll, wird einfach der Jumper-Kontakt (auf der Bestückungsseite an der linken Unterkante des Sockels für EPROM I) mit einem Schraubenzieher

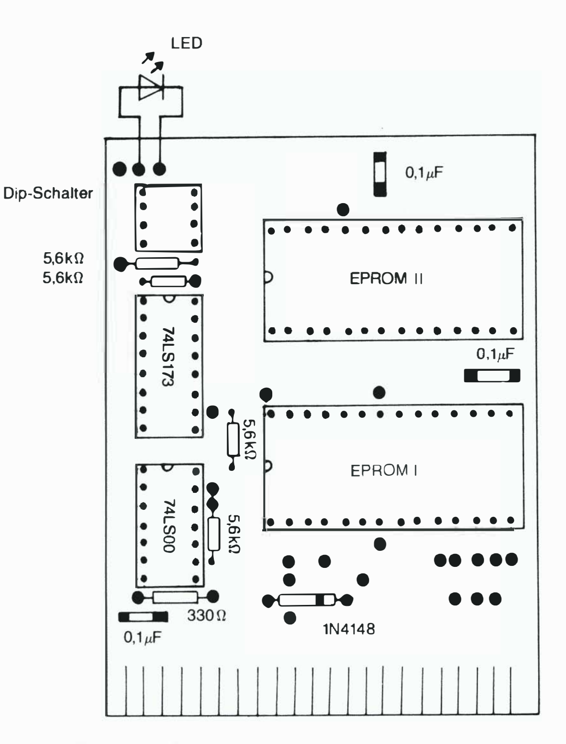


Bild 6.11: Bestückungsplan der 2 x 64-K-EPROM-Karte

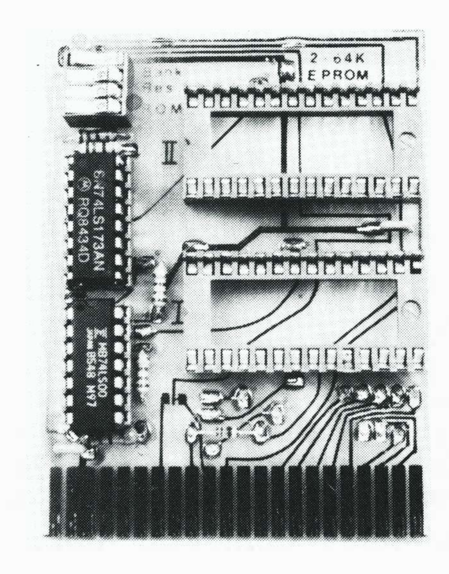


Foto 6.1: Die betriebsbereite EPROM-Karte

durchgetrennt und mittels einer Lötbrücke zur linken Seite gezogen.
Foto 6.1 zeigt die fertig aufgebaute Platine.

6.5 Der Anschluß von Standardbausteinen an den C64

Um möglichst effektiv arbeiten zu können, benutzt der C64 sein internes Bussystem gleich doppelt. Zum einen werden wie üblich Daten zwischen Speichern und CPU hin- und hertransportiert. Wie bereits die Zeitdiagramme in den Bildern 6.6 und 6.7 zeigen, laufen diese Vorgänge vorzugsweise in der zweiten Takthälfte ($Ø_2$ = HIGH) ab. Diese Tatsache macht sich der Videocontroller des C64 zunutze und holt sich seine zum Bildaufbau benötigten Daten immer in der ersten Takthälfte ($Ø_2$ = LOW) aus dem Speicher. Er arbeitet also im Wechsel mit dem Prozessor.
Aus Kapitel 5.1 wissen wir, daß zum Bildaufbau Zeile für Zeile und Punkt für Punkt an den Videoteil gesandt werden müssen. Bei jedem Takt holt sich der Videocontroller 1 Byte und sendet es Bit für Bit an Videoausgang und HF-Modulator. Da 1 Byte aus 8 Bit besteht, erfolgt die Ausgabe mit der achtfachen Taktfrequenz, in der deutschen Version des C64 also mit 7,83 MHz. Diese Frequenz steht als DOTCLOCK am Expansion-Port zur Verfügung.
Das Prinzip der zweifachen Busnutzung ist zwar einleuchtend, bringt jedoch auch gewisse Zeitprobleme mit sich. Normalerweise zeigt den I/O-(Input-/Output- = Eingabe-/Ausgabe-)Bausteinen die steigende Flanke des Taktsignals $Ø_2$ an, daß bereits alle Adreßleitungen den richtigen Pegel für den folgenden Speicherzugriff

haben. Beim C64 wird aber genau zu diesem Zeitpunkt erst zwischen Videocontroller und Prozessor umgeschaltet. Die gültige Adreßkombination liegt erst rund 100 ns nach der positiven Taktflanke mit Sicherheit stabil an.

Alle im C64 eingesetzten Bausteine (zum Beispiel die CIAs 6526) sind eigens für diesen speziellen Zeitablauf konstruiert. Leider muß man solche Besonderheiten im wahrsten Sinne des Wortes teuer bezahlen. Es handelt sich nämlich um Spezialbausteine, die nur von der Commodore-Tochterfirma MOS hergestellt werden und etwa doppelt so viel kosten, wie vergleichbare Standard-ICs.

Mit einem kleinen Trick lassen sich aber auch gewöhnliche 65XX-ICs anderer Firmen an den C64 koppeln. Es ist nur die positive Flanke von Ø2 um eine geeignete Zeit zu verzögern. Bild 6.12 zeigt eine praktikable Schaltung mit dem D-Flipflop 7474.

Die Funktion eines Flipflop wurde bereits in Kapitel 2.3.6.2 detailliert beschrieben. Nochmal zur Erinnerung: Bei einer positiven Flanke an seinem Anschluß CLOCK wird der Zustand des Dateneingangs D an den Ausgang Q übernommen. Der hier verwandte Typ 7474 enthält zusätzlich noch einen Reset-Anschluß $\overline{\text{RES}}$, der bei LOW-Pegel die beiden Ausgänge sofort in ihre Normallage Q = LOW und $\overline{\text{Q}}$ = HIGH bringt.

Zur Triggerung des Flipflop wird die oben erwähnte Bildpunktfrequenz DOT-CLOCK verwendet. Die Eingänge D und $\overline{\text{RES}}$ werden beide mit Ø2 verbunden. In der ersten Taktphase (Ø2 = LOW) bleibt also Q immer LOW. Geht Ø2 jedoch nach HIGH, dann folgt der Ausgang erst nach der nächsten positiven DOTCLOCK-Flanke. Bei der DOTCLOCK-Frequenz von 7,83 MHz ergibt sich eine Verzögerung des Taktsignals Ø2 um etwa 128 ns.

Dabei geht man davon aus, daß Ø2 und

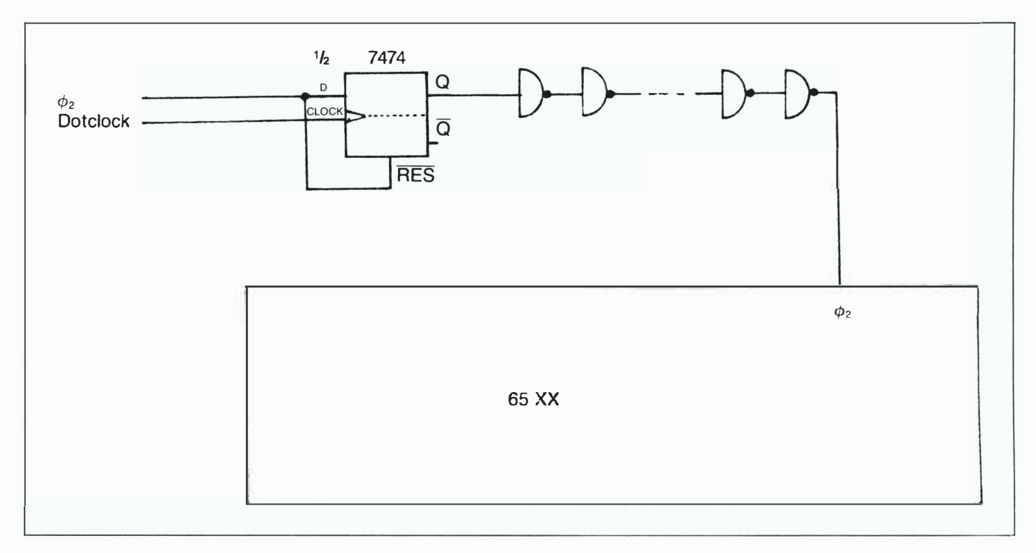


Bild 6.12: Anschaltung eines gewöhnlichen Peripheriebausteins an den C64

DOTCLOCK phasengleich sind, das heißt, daß bei einem Pegelwechsel von Ø2 gleichzeitig auch DOTCLOCK seinen Zustand ändert. Leider hat sich gezeigt, daß es auch Ausführungen des C64 gibt, bei denen sich die DOTCLOCK-Phase bei Erwärmung des Gerätes nach einiger Betriebszeit (je nach Gerät zirka 10 Sekunden bis 30 Minuten) deutlich verschiebt. Es handelt sich dabei vor allem um Rechner jüngeren Datums, in der Regel mit grauen Funktionstasten.
Um auf jeden Fall Funktionssicherheit zu gewährleisten, ist dem Flipflop nachfolgend eine geringfügige Verzögerung einzubauen. Am einfachsten wird das durch mehrere hintereinandergeschaltete Inverter erreicht. Wie Sie wissen, hat jedes Gatter eine bestimmte Verzögerungszeit, die vom Anlegen des Signals am Eingang bis zur Änderung des Ausgangs vergeht. Typischerweise beträgt sie etwa 10 ns. Bisher war diese Verzögerungszeit eigentlich nur störend. Hier kann sie endlich einmal positiv ausgenutzt werden. Bei der Ermittlung der Gatteranzahl muß man eventuell etwas experimentieren, da die Phasendifferenz von Rechner zu Rechner und die Gatterlaufzeit von Chip zu Chip verschieden sein kann. Bewährt hat sich die Hintereinanderschaltung aller sechs Inverter eines 74LS04. Geringfügig längere Zeiten sind mit dem pinkompatiblen CMOS-IC 4069 erreichbar. Damit das verzögerte Taktsignal aber nicht invertiert wird, dürfen nur gerade Gatterzahlen eingesetzt werden.
Mit der beschriebenen Schaltung wird zum Beispiel die Anschaltung des gängigen I/O-Bausteins VIA 6522 möglich.

6.6 Analog/Digital-Wandlung

Bereits in Kapitel 2.2.11 haben Sie ein Gerät zum Umsetzen von Bitkombinationen in eine Spannung kennengelernt. Hier folgt nun das Gegenstück: ein Zusatz, der Spannungen in Bitkombinationen umwandelt – und das sehr schnell. Auf der Platine ist außerdem ein empfindlicher Tonfrequenzvorverstärker enthalten.

6.6.1 Die sukzessive Approximation

Lassen Sie sich von diesem Zungenbrecher nicht einschüchtern. Das Verfahren an sich ist ganz einfach. Bevor wir aber in die Materie einsteigen, sei noch darauf hingewiesen, daß auch die digitalen Meßwertaufnehmer aus Kapitel 2.2.5 (kodierte Lichtschrankenscheibe) als Analog/Digital-Umsetzer aufgefaßt werden können, denn sie erfassen eine analoge Größe – zum Beispiel Winkel, Weg, Drehzahl – und geben ein digitales Signal ab, sind also bereits mechanisch-elektrische Analog/Digital-Umsetzer. In den nachfolgend beschriebenen Schaltungen wird demgegenüber davon ausgegangen, daß das analoge Signal in Form einer Spannung vorliegt. Es handelt sich damit gleichzeitig um sogenannte Digital-Voltmeter, also Geräte zum Erfassen analoger Spannungen mit digitaler Anzeige. Doch zurück zur eigentlichen Umsetzung.
Erinnern wir uns an die Digital/Analog-Wandlung aus Kapitel 2.2.11. Die zugehörigen Spannungen aller gesetzten Bits wurden einfach addiert und schon stand der Wert fest. So leicht ist die Sache hier nicht, denn zwischen den mit fester Bitzahl erreichbaren Werten liegen immer

unendlich viele Zwischengrößen. Man ermittelt daher den genauesten Digitalwert, indem man sich ihm stufenweise (sukzessive) annähert (approximiert). Die Eingangsspannung wird mit einer festen Referenzspannung verglichen und nacheinander für jedes Bit entschieden, ob es gesetzt werden muß oder nicht. Vergleichbar ist diese Annäherung etwa der Suche in einem Telefonbuch. Sie schauen sich die Buchstabenkombination oben auf der Seite an und entscheiden, ob Sie noch eine Seite weiterblättern müssen oder nicht.
Das Verfahren der sukzessiven Approximation kommt mit relativ wenig Aufwand aus und ist recht schnell. Umsetzer nach diesem Verfahren werden als komplette ICs angeboten, so auch der ZN 427 von Ferranti, der im nächsten Abschnitt zum Einsatz kommen soll.

6.6.2 Der ZN 427 als A/D-Wandler

Auf den ersten Blick ein ganz gewöhnliches IC. Der ZN 427 ist ein recht schneller Komparator, der bei einer Taktfrequenz von 1 MHz weniger als 10 Mikrosekunden für die Ermittlung eines vollständigen 8-Bit-Wertes benötigt. Er enthält eine präzise, temperaturstabile Referenzspannungsquelle, ist sowohl TTL- als auch CMOS-kompatibel und besitzt als besonderen Clou eine komplette Mikrocomputer-Schnittstelle. Allerdings benötigt er zum Betrieb eine zusätzliche negative Hilfsspannung.
Schauen wir zunächst das Blockschaltbild des ZN 427 (Bild 6.13) näher an. Pin 6 ist der Eingang für die analoge Meßspannung. Zwischen Pin 8 und 9 (Masse) liegt die Referenzspannungsquelle, die etwa so

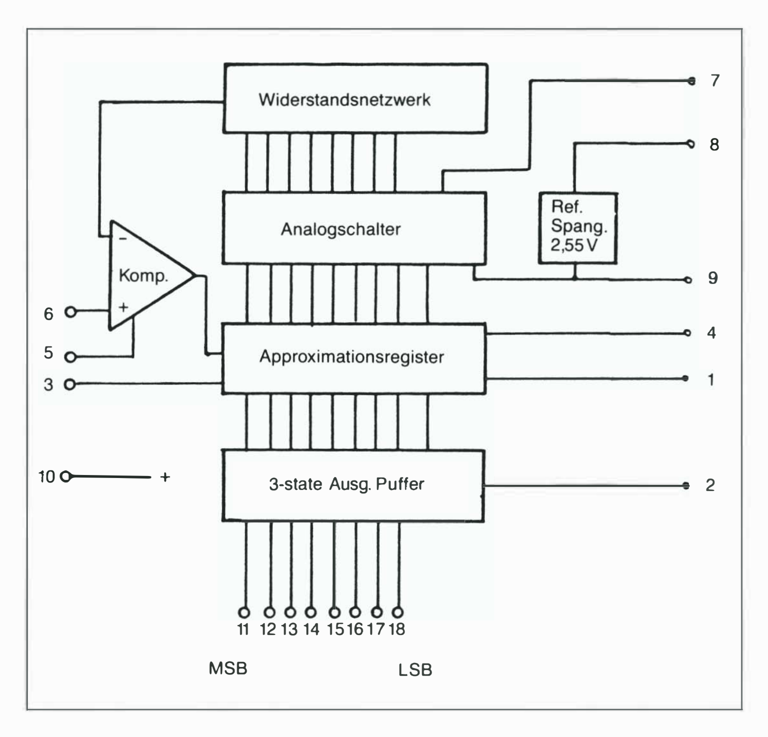


Bild 6.13: Blockschaltbild des A/D-Wandlers ZN 427 von Ferranti

anzusehen ist wie eine Zenerdiode, und mit einem Widerstand nach Pin 10 (+5V) belastet werden muß. Über Pin 7 kann dem IC eine externe Vergleichsspannung zugeführt werden. Im Normalfall sind beide Anschlüsse kurzgeschlossen. Der Meßbereich beträgt dann ± 2,56 Volt.

Bei einer negativen Flanke an seinem Pin 4 ($\overline{WR}$) startet der ZN 427 die Wandlung, indem er Pin 1 ($\overline{BUSY}$) LOW macht und intern zunächst alle Datenbits auf einen Anfangszustand setzt. Sobald $\overline{WR}$ wieder nach HIGH geht, erfolgt nacheinander bei jedem Taktimpuls die Ermittlung eines Bit, wobei das höchstwertige zuerst gültig ist. Nach neun Taktzyklen geht $\overline{BUSY}$ wieder nach HIGH und zeigt damit das Ende der Wandlung an.

Beim Auslesen der Daten verhält sich der ZN 427 ähnlich wie ein EPROM. Seine Datenausgänge besitzen nämlich Tristate-Charakteristik, das heißt, sie lassen sich mit dem Anschluß 2 (RD) ein- und ausschalten. Dieser Anschluß entspricht dem EPROM-Eingang $\overline{OE}$, verhält sich aber hier HIGH-aktiv, das heißt, im Gegensatz zu EPROMs sind die Datenausgänge also bei HIGH durchgeschaltet und bei LOW in einem hochohmigen Zustand, der den Datenbus nicht belastet. So kann der ZN 427 mit sehr wenigen zusätzlichen Gattern direkt an den Expansion-Port des C64 angeschlossen werden.

6.6.3 Die Wandlerkarte am C64

Bild 6.14 zeigt den Schaltplan einer A/D-Wandlerkarte mit dem ZN 427 für den Expansion-Port des C64. Dabei wurde das Interface zum Rechner möglichst einfach gestaltet. Über einen Jumper kann gewählt werden, in welchem Bereich die Karte angesprochen werden soll. Um eine Zusammenarbeit mit anderen Erweiterungen aus diesem Buch zu gewährleisten, sollte jedoch $\overline{I/O1}$ eingestellt sein. Dieses Signal ist nur im Adreßbereich $DE00 bis $DEFF (56 832 bis 57 087) LOW. Es wird über ein NOR-Gatter mit dem invertierten R/$\overline{W}$-Signal verknüpft. Der Ausgang des NOR-Gatters kann also nur HIGH sein, wenn aus einer der genannten Adressen gelesen wird. Über RD (Pin 2) werden bei einer solchen Aktion die Datenausgänge D_0 bis D_7 durchgeschaltet.

Es hat wenig Zweck, immer nur den gleichen Wert auszulesen, denn das geschieht unweigerlich, wenn man nicht zwischendurch mal eine Wandlung startet. Den dazu nötigen Impuls an $\overline{WR}$ (Pin 4) liefert ein Schreibbefehl auf eine der selektierten Adressen. Bei einer Schreiboperation ist ja die R/$\overline{W}$-Leitung des Prozessors LOW. Kommt gleichzeitig auch vom I/O-Jumper ein negatives Signal, dann wird der Ausgang des unteren NOR-Gatters HIGH und der nachgeschaltete Inverter liefert an $\overline{WR}$ das nötige LOW-Signal.

Eine Wandlung sieht dann in BASIC etwa so aus:

```
10 POKE 56832, 0:REM
WANDLUNG STARTEN
20 PRINT PEEK(56832): REM
WERT ABHOLEN
```

Welcher Wert von Zeile 10 in die Adresse geschrieben wird, ist belanglos, denn er wird nirgends gespeichert und steuert auch nichts. Wegen der vergleichsweise langsamen Arbeitsgeschwindigkeit der Interpretersprache BASIC kann getrost auf eine Warteschleife zwischen dem Start der Wandlung und dem Abholen des Ergebnis-

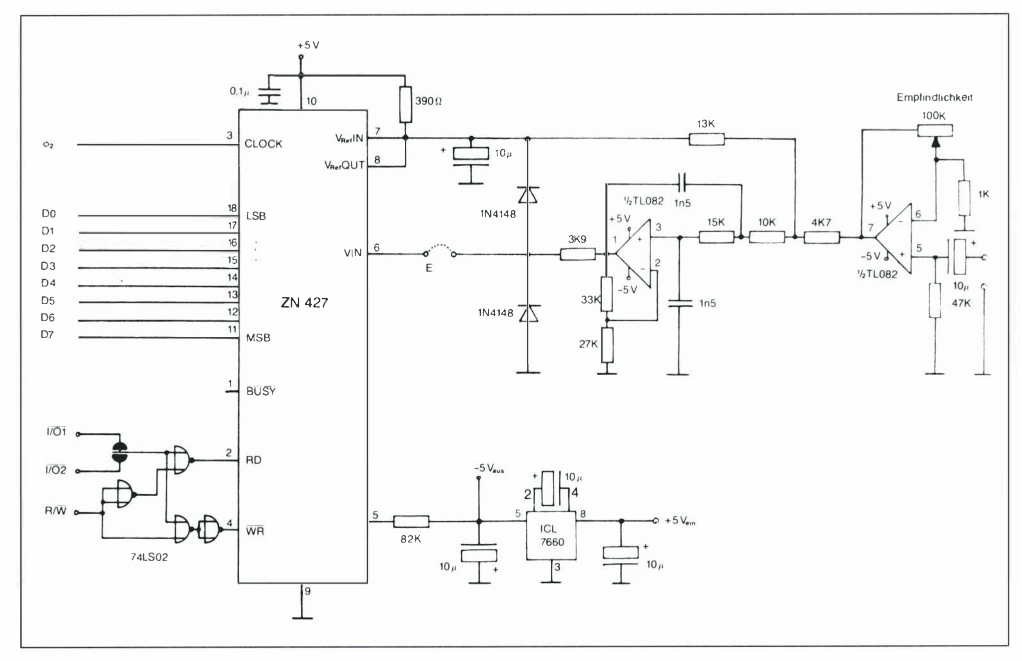


Bild 6.14: Schaltplan des A/D-Wandlers

ses verzichtet werden. Anders liegt die Sache schon bei Betrieb in Maschinensprache.

Der Wandler ist schnell genug zur Abtastung von Tonfrequenzsignalen. Daher wurde gleich ein entsprechender Verstärker auf der Platine untergebracht. Er wird mit einer steckbaren Drahtbrücke an das Wandler-IC angekoppelt. Soll der ZN 427 zur Messung von Gleichspannungen dienen, kann bei entfernter Drahtbrücke der Eingang E benutzt werden. Die Bilder 6.15 und 6.16 geben Meßschaltungen für unterschiedliche Bereiche an. Dabei beziehen sich die Widerstandswerte in Klammern jeweils auf den größeren Meßbereich (+10 Volt bzw. ±10 Volt).

Die relativ aufwendige Verstärkerschaltung auf der Platine dient nicht nur zur Anpassung des Pegels, sondern erfüllt gleichzeitig noch eine weitere wichtige Aufgabe. Beim Wandlungsverfahren der sukzessiven Approximation darf sich die Eingangsspannung während der Umsetzung nicht ändern, da sonst die ganze Theorie durcheinandergeraten und falsche Ergebnisse ermittelt würden. Pfeifen und Zwitschern wären die Folge. Daher ist ein mehrstufiger Tiefpaß enthalten, der alle Frequenzen oberhalb von etwa 10 kHz unterdrückt. Vor dem Tiefpaß liegt noch ein Vorverstärker, dessen Empfindlichkeit in einem weiten Bereich mit einem Poti einstellbar ist. Auf diese Weise sind sogar Mikrofone direkt anschließbar. Die Referenzspannung des Wandlers wird über einen Spannungsteiler in die Filterstufe eingespeist. So stellt sich die Anordnung selbsttätig auf einen Wert

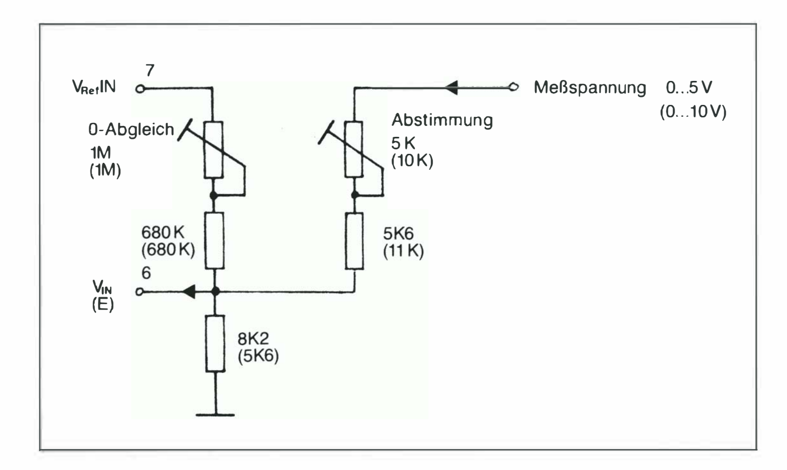


Bild 6.15: Zusatzschaltung zum Messen positiver Spannungen

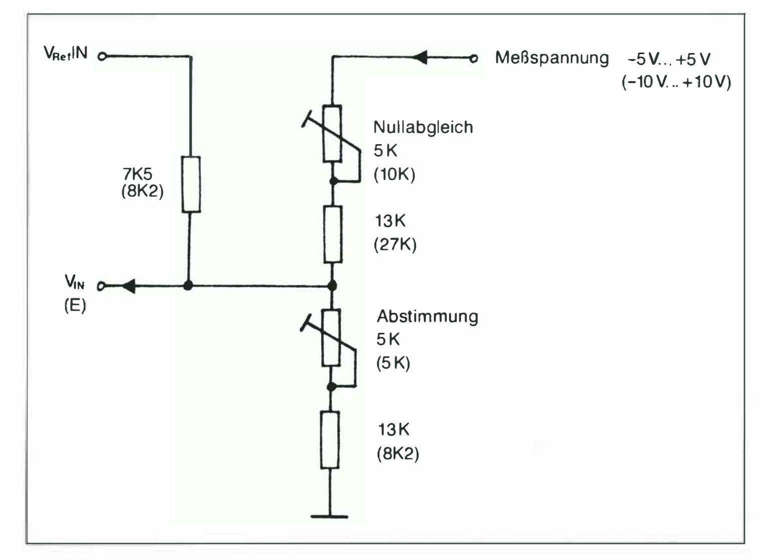


Bild 6.16: Zusatzschaltung zur Erfassung symmetrischer Meß-spannungen mit dem ZN 427

in der Mitte des 8-Bit-Bereiches ein, und es ist gewährleistet, daß beide Halbwellen des Tonsignals gleich gut verarbeitet werden. Der maximale Eingangspegel liegt damit bei etwa $3V_{ss}$, also bei etwa $1V_{eff}$.
Die nötige negative Hilfsspannung von -5 Volt für Operationsverstärker und Wandler wird mit einem Spannungsumsetzer vom Typ ICL 7660 erzeugt. Leider gibt es am Expansion-Port keine Wechselspannung, die zum Erzeugen einer zusätzlichen Spannung benutzt werden kann. Der ICL 7660 wird im Netzteil-Kapitel 7.2 noch genauer vorgestellt.
Seinen Takt erhält der ZN 427 direkt aus dem Systemtakt $Ø_2$ des C64. Das hat den Vorteil, daß die Umsetzzeit einer Reihe genau abzählbarer Programmschritte entspricht und so in Maschinenprogrammen keine Wartezeit verloren zu gehen braucht.

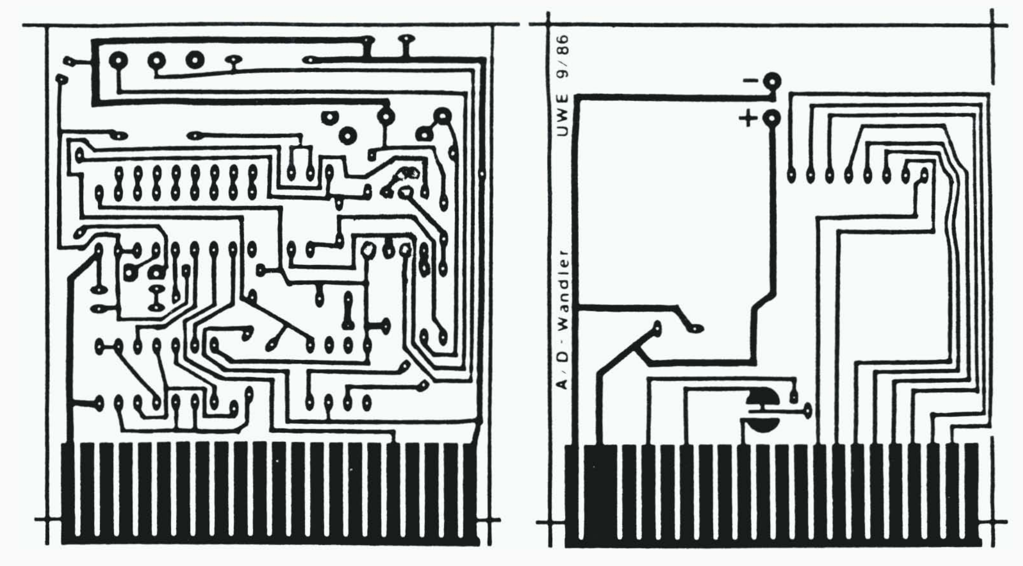


Bild 6.17: Platinenlayout für die A/D-Wandlerkarte

6.6.4 Aufbau des A/D-Wandlers

Bild 6.17 zeigt die doppelseitig ausgeführte Platine des Analog/Digital-Wandlers. Obwohl ihr Schaltplan (Bild 6.14) recht unscheinbar aussieht, ist sie dicht bepackt und enthält vor allem viele Leiterbahnen. Richten Sie sich beim Herstellen der Platine und beim Bestücken (Bild 6.18) nach den Hinweisen im Anhang A. Tabelle 6.8 gibt die benötigten Bauteile an.

Die fertige Platine ist auf Foto 6.2 zu sehen. Sie wird direkt in den Expansion-Port gesteckt und benötigt daher auf dieser Seite keinen Steckverbinder. Achten Sie bei der Platinenherstellung unbedingt auf das Einhalten der richtigen Breite im Steckerbereich.

Die DIN-Buchse kann bei Bedarf auch durch andere Ausführungen bzw. durch Lötnägel ersetzt werden. Allerdings sollte die Zuleitung abgeschirmt ausgeführt sein,

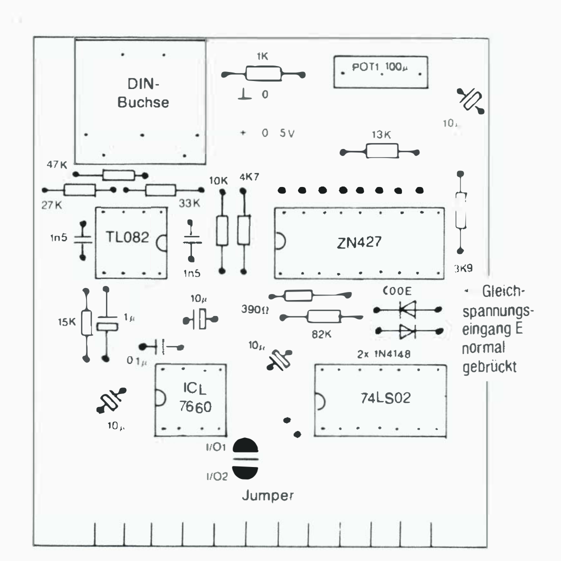


Bild 6.18: Bestückung der A/D-Wandlerkarte

wobei das Geflecht an Masse liegen muß. Für spezielle Zwecke ist an zwei Lötnägeln die Versorgungsspannung von +5 Volt und

1 A/D-Wandler ZN 427E-8
1 Doppel-Operationsverstärker TL 082
1 DC/DC-Wandler ICL 7660
1 Logik-Baustein 74LS02
1 IC-Sockel 18polig
1 IC-Sockel 14polig
2 IC-Sockel 8polig
2 Dioden 1N4148
1 Widerstand 82 Kiloohm
1 Widerstand 47 Kiloohm
1 Widerstand 82 Kiloohm
1 Widerstand 33 Kiloohm
1 Widerstand 27 Kiloohm
1 Widerstand 15 Kiloohm
1 Widerstand 13 Kiloohm
1 Widerstand 10 Kiloohm
1 Widerstand 4,7 Kiloohm
1 Widerstand 3,9 Kiloohm
1 Widerstand 1 Kiloohm
1 Widerstand 390 Ohm
1 Potentiometer 100 Kiloohm, logarithmisch
5 Elektrolytkondensatoren 10 Mikrofarad, radial
1 Kondensator 0,1 Mikrofarad, Keramik
2 Kondensatoren 1.5 Nanofarad
1 DIN-Buchse, 5polig, direkt einlötbar
4 Lötstifte
1 doppelseitige Platine nach Bild 6.17

Tabelle 6.8: Einkaufszettel für den A/D-Wandler

Masse herausgeführt. Diese Lötnägel sind vorsichtig von der Platinenoberseite her zu verlöten.

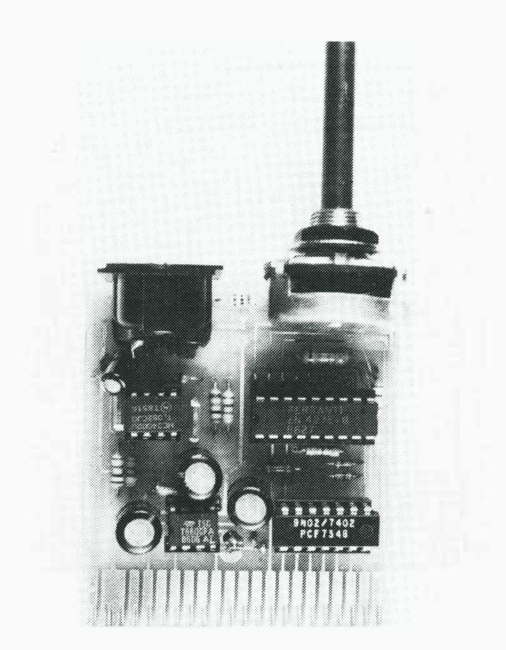


Foto 6.2: Der A/D-Wandler

6.6.5 Der C64 als Digital-Hallgerät

Sowohl die A/D-Wandlerkarte, wie auch der D/A-Wandler aus Kapitel 2.2.11 sind für Tonfrequenzanwendungen vorbereitet. Durch eine einfache Schleife wird der C64 dazu veranlaßt, für kurze Zeit Wert auf Wert in seinen Datenranzen zu fressen. Der so abgespeicherte Spannungsverlauf kann anschließend beliebig oft abgespielt werden. Sicherlich kennen Sie bereits solche digitalisierte Musikpassagen, bei denen meist der Sound-Chip des C64 mit seinen vier Lautstärkestufen zur Ausgabe benutzt wird. Unsere beiden Karten ermöglichen jedoch eine Quantifizierung zu 8 Bit, das heißt, es lassen sich 256 verschiedene Spannungen erzeugen. Sie werden über die Qualität der Ausgabe erstaunt sein!

Den Rechner als Recorder zu benutzen, mag zunächst ganz interessant erscheinen,

doch diese Methode ist äußerst speicherintensiv. Ohne Schwierigkeiten könnten Sie innerhalb kürzester Zeit einige Gigabyte mit Musik oder Sprache füllen. Angesichts dieser Tatsache bleibt man wohl doch lieber beim guten alten Kassettenrecorder.

Durch die beiden Wandlerkarten und das Programm DIGITALHALL sind Sie in der Lage, mit dem C64 Echo- und Halleffekte zu generieren. Stecken Sie dazu die A/D-Karte in den Expansion-Port und schließen Sie an seiner DIN-Buchse ein Mikrofon an. Ebenso stecken Sie die D/A-Karte auf den User-Port und führen dessen Ausgangssignal zu einem Verstärker. Nun schalten Sie den Rechner ein, laden das Programm DIGITALHALL.OBJ und starten es mit SYS 12*4096 ($C000).

Im Lautsprecher werden Sie dann die Geräusche am Mikrofon um beträchtliche Zeit verzögert hören. Wenn Sie RESTORE drücken, nimmt der Rahmen eine helle Farbe an und die Nachhalldauer wird symbolisch durch eine Zahl angezeigt. Sie befinden sich nun im Einstellmodus und können diesen Wert mittels der Tasten »+« und »-« ändern. Sobald Sie RETURN drücken, gelangen Sie wieder in den Funktionsmodus. Durch Drücken von Q im Einstellmodus wird das Programm beendet.

Wenn Sie eine kleine Zahl einstellen, und das Mikrofon beim Sprechen dicht an den Lautsprecher halten, werden Sie den Halleffekt bereits deutlich registrieren.

Das Programm besteht aus drei Teilen: der Initialisierung, der Hauptschleife und einer geänderten NMI-Routine. Während der Hallerzeugung in der Hauptschleife bleibt die IRQ-Behandlung gesperrt, um möglichst wenig Zeit verlorengehen zu lassen. Gleichzeitig ist dadurch auch die Tastaturabfrage unterbunden. Durch Drücken von RESTORE wird ein nicht maskierbarer Interrupt ausgelöst, der die NMI-Routine startet. Lesen und Speichern geschieht zyklisch über zwei Vektoren in der Zero-Page, nämlich POINTER1 / POINTER1+1 und POINTER2 / POINTER2+1, die in ihrer Phase um die Verzögerungszeit jeweils in Schritten zu 256 Byte versetzbar sind. Der benutzte RAM-Bereich liegt zwischen MEMMIN und MEMMAX, läßt sich also leicht durch Ändern der Konstanten variieren. Angegeben ist jeweils das Highbyte. Beachten Sie jedoch, daß der gesamte Block des angegebenen Byte noch mitbenutzt wird. Bei MEMMAX = $9F sind alle Speicherzellen bis $9FFF belegt.

6.7 Betriebssystem, wechsle dich!

6.7.1 PROM oder EPROM?

In jedem Mikrocomputersystem befindet sich Firmware. So nennt man den fest in ROMs untergebrachten Speicherinhalt, der entweder das Betriebssystem bzw. die Programmiersprache beherbergt, einen Zeichensatz darstellt oder einfach den Urlader enthält, wenn das eigentliche System erst von Diskette gebootet werden soll. In jedem Fall verwendet man Speicherbausteine, die auch nach Abschalten der Betriebsspannung ihren Inhalt nicht vergessen: Read Only Memories – ROMs.

Im C64 versehen zwei Festwertspeicher mit der Typenbezeichnung 2564 ihren Dienst. Sie enthalten in jeweils 8 Kilobyte den BASIC-Interpreter (U3) und das Betriebssystem (Kernel: U4). Für spezielle Zwecke

würde mancher Anwender gerne einzelne Routinen aus dem darin abgelegten Programm ändern. Da es sich aber bei Bausteinen mit der Bezeichnung 25XX um nicht löschbare maskenprogrammierte PROMs handelt, wird der betroffene Speicher durch ein selbstprogrammiertes EPROM ersetzt. Wie die einzelnen Pins im Fall des 2564 und 2764 zu verbinden sind, wird in Tabelle 6.9 gezeigt.

EPROM-Sockel 28polig		Original-Sockel 24polig 2564
1,28,27,26	———	24
2	———	21
3	———	1
4	———	2
5	———	3
6	———	4
7	———	5
8	———	6
9	———	7
10	———	8
11	———	9
12	———	10
13	———	11
14,20	———	12
15	———	13
16	———	14
18	———	15
18	———	16
19	———	17
21	———	19
22	———	20
23	———	18
24	———	22
25	———	23

Tabelle 6.9: Verbindungen zwischen den beiden Sockeln

6.7.2 Nicht nur für den C 64

Die bisher gebräuchlichste PROM-Größe ist 8 Kilobyte organisiert zu 8 Bit, also der Baustein 2564. Er taucht in sehr vielen Geräten auf, beispielsweise auch in der zum C 64 gehörenden Floppystation VC 1541, in verschiedenen Druckern und außerdem in diversen kommerziellen Geräten. Die in Bild 6.19 gezeigte kleine Platine ermöglicht es, den Originalbaustein sowie bis zu vier zusätzliche Speicherbereiche eines EPROM gleichzeitig gegeneinander umschaltbar in einem Sockel unterzubringen. Die Umschaltung kann dabei jeweils mit einem einfachen Schalter erfolgen, oder – bei geeigneter Programmierung – auch über eine Portleitung per Software direkt vom Computer aus. Durch die Möglichkeit, jederzeit die Original-Firmware parat zu haben, werden Kompatibilitätsprobleme zu bereits vorhandener, alter Software von vornherein ausgeschlossen. Dabei ist die einseitig kupferkaschierte Platine so klein gehalten, daß sie in alle Geräte auch bei akutem Platzmangel eingesetzt werden kann.

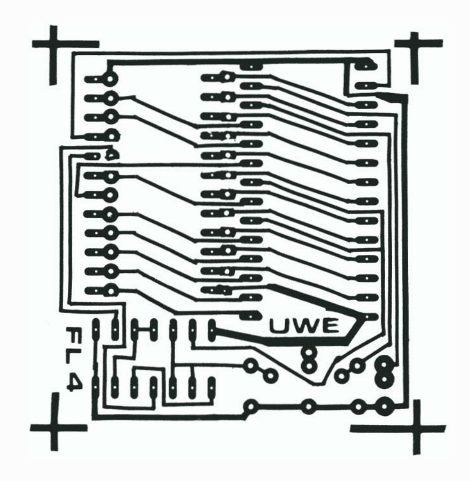


Bild 6.19: Platinenvorlage für einen Firmwareumschalter

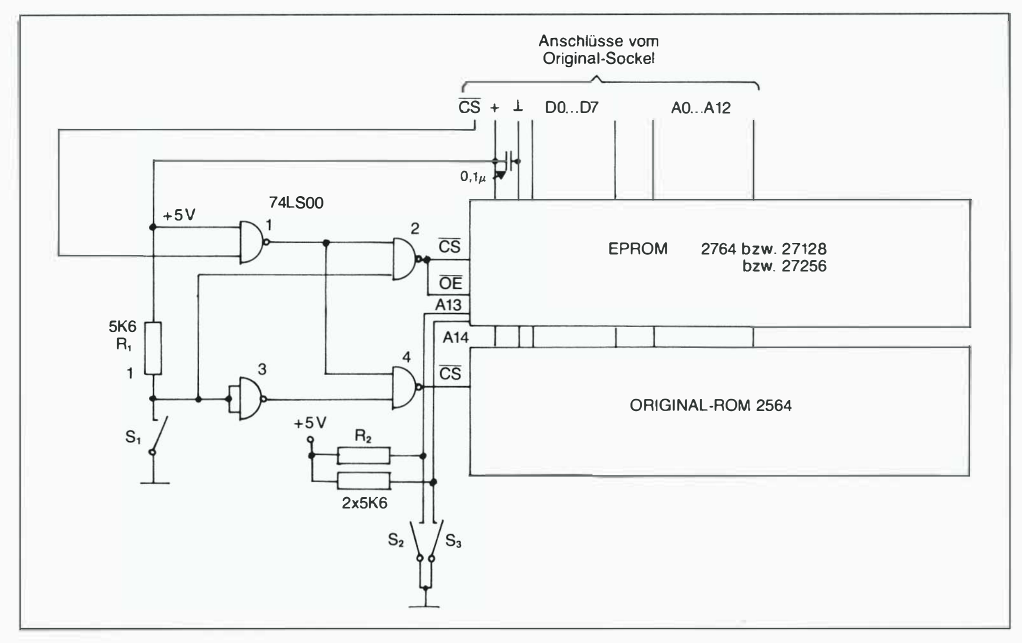


Bild 6.20: Schaltbild des Firmwareumschalters

6.7.3 Klein und alternativ

In Bild 6.20 ist der Schaltplan der Hilfsplatine wiedergegeben. Um den Hardwareaufwand möglichst gering zu halten, und um Einsetzbarkeit in möglichst allen Geräten – zum Beispiel auch der Floppy – garantieren zu können, wurde die Platine nicht für irgendeine feste gerätespezifische Steckverbindung geplant, sondern es wurde einfach der vorhandene EPROM-Sockel ausgenutzt. An ihm sind bereits alle benötigten Leitungen herausgeführt. Zur Steuerung wird lediglich ein TTL-IC 74LS00 mit vier NAND-Gattern benötigt. Über Kontaktleisten, die einseitig in die Platine gelötet sind und mit ihrem freien Ende in die Original-EPROM-Fassung auf der Hauptplatine gesteckt werden, verbindet man alle korrespondierenden Pins miteinander – außer der Chip-Select-Leitung $\overline{CS}$. Sie wird auf den Eingang des NAND-Gatters 1 gegeben und steht an je einem Eingang von Gatter 2 und Gatter 4 invertiert zur Verfügung. Mit Hilfe des Schalters S_1 und des Pull-up-Widerstands R_1 wird der andere Eingang des Gatters 2 mit HIGH-Pegel bei offenem bzw. LOW-Pegel bei geschlossenem Schalter beschickt.

Der zweite Eingang am NAND-Gatter 4 verhält sich genau komplementär dazu, da hier der Eingangspegel durch Gatter 3 invertiert wird. Somit ist je nach Schalterstellung der Eingang $\overline{CS}$ entweder auf das EPROM oder auf das Original-ROM durchgeschaltet.

Als EPROM kann wahlweise ein 2764 mit 8 Kilobyte, ein 27 128 mit 16 Kilobyte oder ein 27 256 mit 32 Kilobyte Speicherkapazität verwendet werden. Im 2764 findet

ein zusätzliches Betriebssystem umschaltbar gegen das Original Platz, im 27 128 zwei und im 27 256 gar vier.

Beim 27 128 entscheidet der logische Pegel am Adreßanschluß A_{13} darüber, ob der obere oder der untere 8-Kbyte-Bereich ausgewählt ist. Auch hier geschieht die Erzeugung des logischen Pegels über einen Schalter (S_2) und den Pull-up-Widerstand R_2. Ein weiterer Schalter bedient den Anschluß A_{14}, der natürlich nur beim EPROM-Typ 27 256 ausgewertet wird. Findet ein 2764 Verwendung, dann ist die Stellung der Schalter unerheblich.

6.7.4 Aufbau und Inbetriebnahme des EPROM-Adapters

Die Einzelteile – zusammengestellt in Tabelle 6.10 – werden wie im Bestückungsplan (Bild 6.21) auf der Platine nach dem Layout in Bild 6.19 verlötet. Dabei ist darauf zu achten, daß die gestrichelt dargestellten Kontaktleisten von der Platinenunterseite her einzusetzen sind (Foto 6.3). Alle drei Pull-up-Widerstände müssen aus Platzgründen stehend eingebaut werden.

1	Platine nach Bild 6.20
1	Kontaktleiste, mind. 24pol. (z.B. Bürklin DV-Nr. B102.114)
1	IC-Sockel 24pin
1	IC-Sockel 28pin
1	IC-Sockel 14pin
1	74LS00
1	2764 bzw. 27 128 EPROM
3	Widerstände 5,6 kΩ
1	Kondensator 0,1μF, Keramik
3	Schalter 1x ein
ca. 1 m	Verdrahtungslitze

Tabelle 6.10: Stückliste des EPROM-Adapters

Die Zuleitungen zu den Schaltern dürfen zwar beliebige Länge haben, sollten aber nicht sehr viel länger als nötig sein, um überflüssigen Kabelsalat im Gehäuse zu vermeiden. Der Austausch des PROM darf nur bei ausgeschaltetem Gerät vorgenommen werden (Foto 6.4). Beachten Sie auch die Hinweise zur Behandlung von MOS-ICs im Anhang A, denn auch Speicher fallen unter diese Kategorie. Kennzeichnen Sie sich deutlich, auf welcher Seite des IC die Kerbe für Pin 1 liegt. Bei falschem Einsetzen kann ein Bauteil zerstört werden.

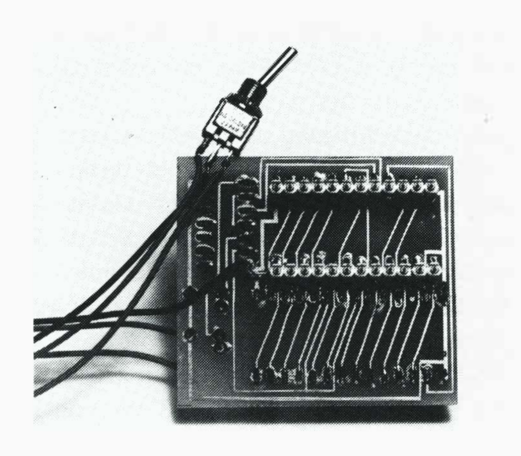


Foto 6.3: Die Umschaltplatine

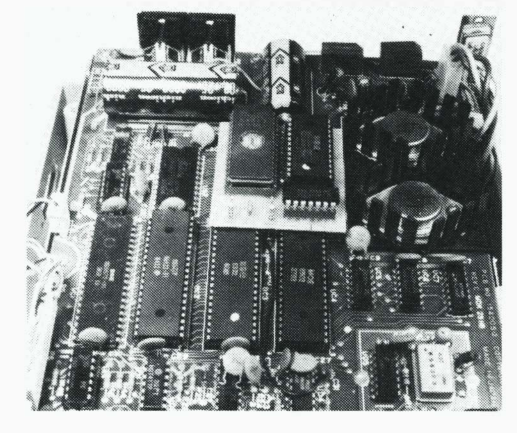


Foto 6.4: Die Platine, eingebaut in eine Floppy VC1541

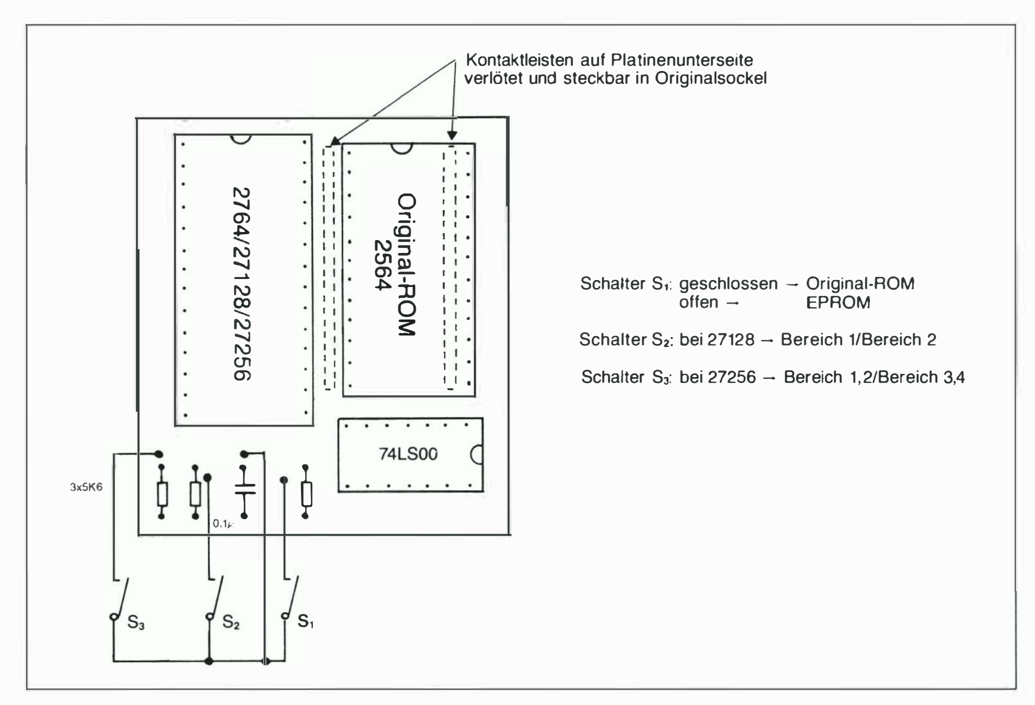


Bild 6.21: Bestückungsplan des EPROM-Adapters

Hebeln Sie nun das Original-IC mit einem Schraubenzieher vorsichtig aus seinem Sockel, und stecken es gleich in die 24polige Fassung auf der bereits mit dem 74LS00 und dem gebrannten EPROM bestückten kleinen Hilfsplatine. Die Kerbe im Gehäuse muß sich dabei dort befinden, wo sie auch im Bestückungsplan (Bild 6.21) markiert ist. Nun wird die Zusatzplatine in die jetzt freie Originalfassung eingesteckt, und zwar so, daß die Kerbe des gerade oben eingesteckten 2564 wieder in dieselbe Richtung zeigt, wie vorher.

Bei geschlossenem Schalter S_1 muß sich das System dann nach dem Einschalten genauso verhalten wie vor dem Eingriff.

Kapitel 7
Die Energieversorgung

Ohne Energie taugt die beste Elektronik nicht viel. Aber mit »irgendeinem« Netzteil ist es meist nicht getan. Die Stromversorgung muß auf den jeweiligen Anwendungsfall zugeschnitten sein.
Dieses Kapitel gibt einen Überblick, wie die Verhältnisse beim C 64 aussehen, zeigt, wie auf Zusatzplatinen leicht auch negative Hilfsspannungen zu erzeugen sind, und erörtert schließlich, welche Verbesserungen an der Original-Netzversorgung des C 64 möglich sind.

7.1 Wozu ein Netzteil?

Über einen Transformator wird die 220-Volt-Wechselspannung aus der Steckdose auf einen für die Regelung brauchbaren Wert herabgesetzt, gleichgerichtet, gesiebt und der jeweils notwendigen Regelung zugeführt. Commodore hat sich beim C 64 dazu entschlossen, das Netzteil in einem eigenen Gehäuse unterzubringen. Sicherlich hat das sowohl Vor- als auch Nachteile. Über einen siebenpoligen DIN-Stecker werden dem C 64 verschiedene Spannungen zugeführt, die wir einmal näher betrachten wollen.
Im Anhang C finden Sie die Pinbelegung des Netzteilsteckers an der rechten Seite des C 64.
Zunächst sind da zwei Wechselspannungsanschlüsse, die mit 9 VAC (AC = Alternating Current = Wechselstrom) bezeichnet sind. Sie werden im Computer über diverse Entstörglieder sowie einen Kontakt des POWER-Schalters und eine Sicherung unter anderem auch an den User-Port geführt. Dabei liegt der Schalter im Zweig für Pin 10 und die Sicherung in der Zuführung zu Anschluß 11. Mißt man an diesen beiden Kontakten mit einem Os-

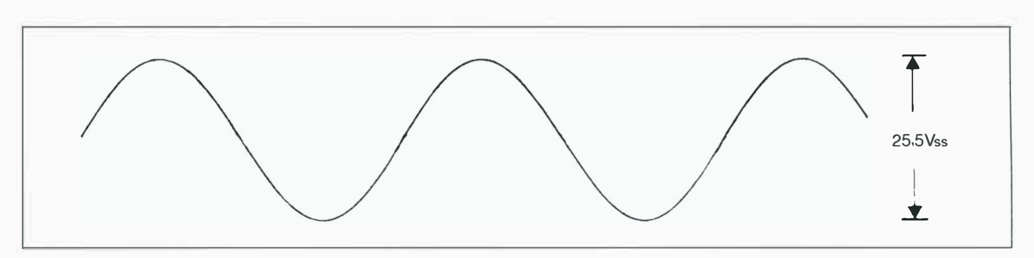


Bild 7.1: Spannungsverlauf an den User-Port-Anschlüssen 10 und 11, gegeneinander gemessen

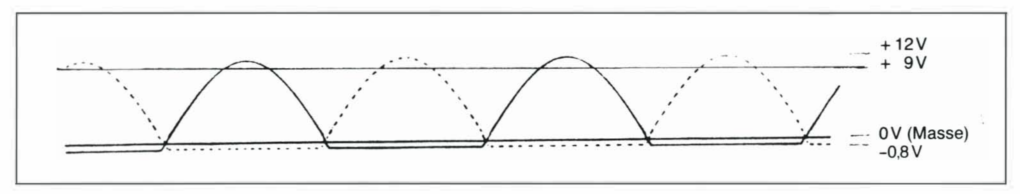


Bild 7.2: Die Spannungsverläufe an den beiden 9-VAC-Anschlüssen 10 und 11 am User-Port auf Masse bezogen

zilloskop, sieht man einen nahezu sinusförmigen Spannungsverlauf nach Bild 7.1, der lediglich durch die kapazitiven und induktiven Einwirkungen der Entstörglieder etwas verzerrt ist.
9 Volt ist der Effektivwert dieser Spannung, das heißt ein Mittelwert, der den Betrag einer Gleichspannung mit derselben Wirkung angibt. Zeitweise werden deutlich höhere Spitzenwerte erreicht, in unserem Fall maximal 25,5 Volt.
Doch Achtung: Die 9-Volt-Anschlüsse sind aufgrund ihrer inneren Verschaltung nicht potentialfrei! Werden sie einzeln gegen Masse gemessen, ergibt sich ein Verlauf nach Bild 7.2. Die Spannung am ersten Pin ist durchgehend, die am anderen gestrichelt gezeichnet. Beide erzeugen abwechselnd eine positive Halbwelle, jeweils mit der Netzfrequenz von 50 Hz.
Neben der Wechselspannung liefert das Netzteil bereits eine geregelte Gleichspannung von +5 Volt. Die Regelung im Netzteilgehäuse hat den Vorteil, daß die dort entstehende beträchtliche Verlustwärme nicht noch zusätzlich den Computer aufheizt.

7.2 Erzeugung von Hilfsspannungen

Es ist kein Zufall, daß die gerade beschriebene Wechselspannung ausgerechnet am User-Port zugänglich ist. Sie läßt sich universell zur Erzeugung diverser Hilfsspannungen verwenden. Dazu werden hauptsächlich Kondensatoren und Dioden verwendet.
Ein Kondensator kann elektrische Energie für kurze Zeit speichern. Er lädt sich schnell bis zur angelegten Spannung auf und gibt im Bedarfsfall Strom an die übrige Schaltung ab, wobei seine Spannung wieder sinkt.
Eine Diode ist ein Halbleiterbauelement, das Strom nur in einer Richtung durchläßt, nämlich nur dann, wenn die Spannung in Richtung des stilisierten Pfeils kleiner ist, als vor dem Pfeil.

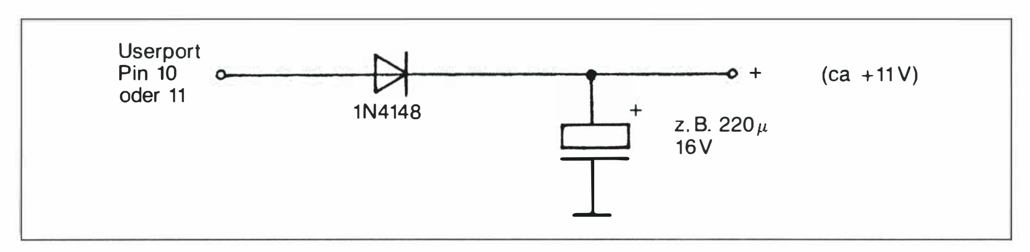


Bild 7.3: Spitzenwertgleichrichter

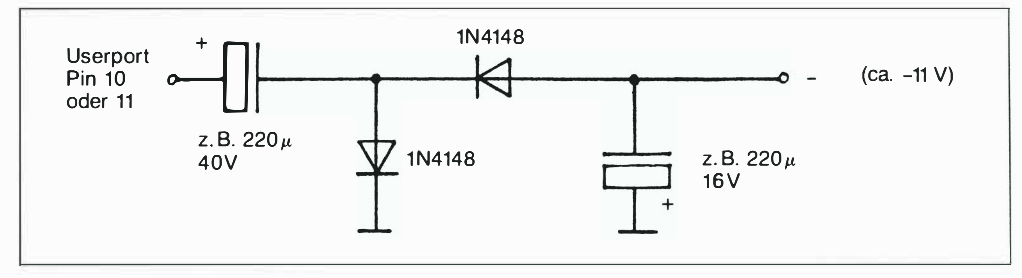


Bild 7.4: Gleichwertabtrennung und Spitzenwertgleichrichter zur Erzeugung negativer Spannungen

Damit dürfte die Anordnung nach Bild 7.3 bereits klar sein. Sie wird von einem Pol der Wechselspannung gespeist. Beim Einschalten ist die Spannung am Kondensator sehr klein. Es kann also Strom durch die Diode fließen und den Kondensator aufladen. Bei sinkender Spannung verhindert die Diode jedoch, daß sich der Kondensator in diese Richtung entlädt. Strom kann nur zum angeschlossenen Verbraucher fließen. Es kommt auf den Bedarf dieses Verbrauchers an, wie weit die Spannung am Kondensator absinkt, bis die nächste Halbwelle ihn wieder aufladen kann. Bei geringem Stromfluß entsteht eine annähernd konstante Gleichspannung in der Größe der maximalen Halbwellenspannung, also von etwa +11 Volt gegen Masse. Eine solche Schaltung wird Spitzenwertgleichrichter genannt.
Mit den an sich positiven Halbwellen ist es auch möglich, negative Spannungen zu erzeugen. Dafür muß zunächst die Wechselspannung potentialfrei, also unabhängig von Masse gemacht werden. In unserem Beispiel geschieht das wieder mit Hilfe eines Kondensators, dessen negativer Pol über eine Diode an Masse gelegt wird. Durch die Vorgänge beim Laden des Kondensators stellt sich an seinem mit den Dioden verbundenen Anschluß eine negative pulsierende Spannung ein, die in ähnlicher Weise wie beim positiven Spitzenwertgleichrichter einen nachgeschalteten Kondensator über eine weitere Diode negativ gegen Masse auflädt. Bild 7.4 zeigt die gesamte Anordnung. Am Ausgang stehen – je nach Wert des Trennkondensators – etwa -11 Volt zur Verfügung.
Beide Schaltungen haben einen Schwachpunkt: Sie sind nur für geringe Belastung sinnvoll einsetzbar. Bei größerem Stromfluß entlädt sich der Kondensator sehr schnell und am Ausgang entsteht ebenfalls eine pulsierende Spannung.
Dieses Problem läßt sich durch einen Brückengleichrichter nach Bild 7.5 beheben. Er besteht aus vier Dioden und wird

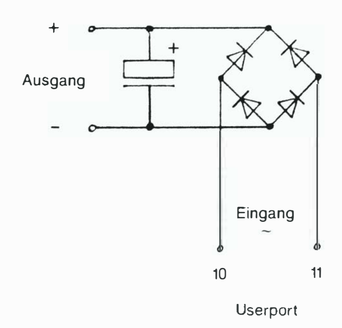


Bild 7.5: Ausnutzung aller Halbwellen mit dem Brückengleichrichter

zwischen die Anschlüsse 10 und 11 des User-Port geschaltet. Ist Pin 10 positiv gegen Pin 11, fließt der Strom durch die obere linke Diode zum positiven Ausgangspol und vom negativen über die untere rechte Diode wieder zurück. Bei der entgegengesetzten Halbwelle ist Pin 11 positiv gegenüber Pin 10. Der Stromfluß erfolgt nun durch die obere rechte Diode zum positiven Ausgang und durch den Verbraucher über die untere linke Diode wieder zurück. Alle anderen Wege sind jeweils gesperrt.

Statt einmal pro Periode, wird der Kondensator jetzt von beiden Halbwellen aufgeladen und kann sich auch bei größeren Strömen nicht sehr weit entladen.

Brückengleichrichter gibt es fertig in Gehäusen mit vier Anschlüssen. Aufgedruckt sind die Maximalwerte für Spannung in Volt und Strom in Milliampere. Die Angabe B80/C1500 bedeutet zum Beispiel maximal 80 Volt und 1,5A.

Schaltet man den negativen Ausgangspol des Brückengleichrichters an Masse, entsteht am positiven Anschluß eine Ausgangsspannung von +10 Volt. Legen Sie aber nie den Pluspol des Gleichrichters an Masse! Aufgrund der Innenschaltung des C64 wäre ein Kurzschluß und das Durchbrennen der Sicherung die Folge.

Manchmal werden höhere Spannungen benötigt, als die bisher erzeugten. Eine Möglichkeit ist dann die Verwendung eines Trafos. Ein Beispiel dafür ist die Erzeugung der Hochspannung zum Betrieb einer Radioaktivitätsmeßröhre in Kapitel 2.4.5.

Als universelle Schaltung zeigt Bild 7.6 eine Kaskade aus Kondensatoren und Dioden. Ihre Funktion beruht darauf, die Kondensatoren parallel aufzuladen und sie dann in Serie zu entladen. So kann man eine um die Anzahl der Kondensatoren vervielfältigte Spannung entnehmen. Der nachgeschaltete Kondensator bildet zusammen mit der letzten Diode wieder einen Spitzenwertgleichrichter.

Bild 7.7 zeigt eine weitere Methode unter Verwendung eines Spezial-IC von Texas, an dessen Ausgang gleich stabilisierte Spannung abgreifbar ist.

Ebenso lassen sich auch negative Spannungen mit Hilfe integrierter Schaltungen generieren. Bild 7.8 zeigt den Spannungswandler ICL 7660, der eine Eingangsspannung von +5 Volt in eine Ausgangsspannung von -5 Volt umkehrt.

7.3 Das Geheimnis stabiler Spannungen

Die Spannungsangaben zu den Schaltungen im vorigen Abschnitt verrieten schon eine gewisse Unsicherheit. Je nach Bela-

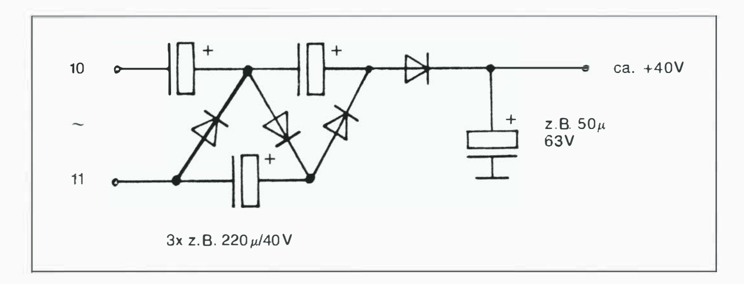


Bild 7.6: Spannungsvervielfachung mit Kaskadenschaltung

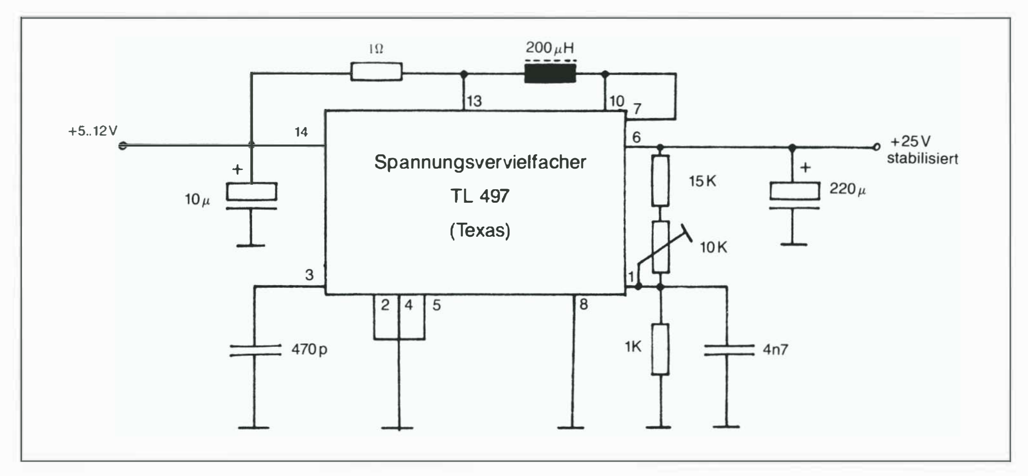


Bild 7.7: Erzeugung einer höheren Gleichspannung

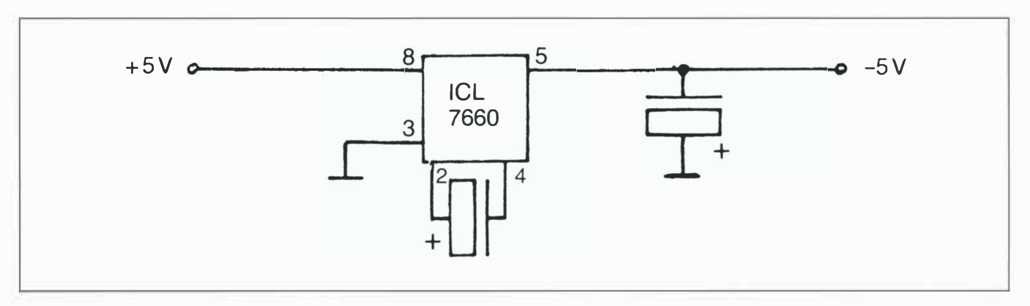


Bild 7.8: Erzeugung einer negativen Spannung mit dem IC ICL 7660

stung können die Werte schwanken. In Kapitel 1 wurde erwähnt, daß gerade digitale TTL-Bausteine nur in einem engen Spannungsbereich zuverlässig arbeiten. Erlaubt sind Schwankungen um ±5%, also liegen die Grenzen der Betriebsspannung bei +4,75 Volt und +5,25 Volt. Die Schwankungen bei größerer Stromentnahme können schnell den zulässigen Minimalwert unterschreiten.

Lastabhängige Spannungsschwankungen können durch einen elektronischen Regler unterdrückt werden, der die Spannung unter allen Umständen konstant zu halten versucht. Heute werden zu diesem Zweck viele integrierte Schaltungen in transistorähnlichen Gehäusen angeboten. Bild 7.9 zeigt eine Standardschaltung mit dem Reglertyp 78XX. Die Buchstaben XX stehen dabei für die gewünschte Höhe der stabilen Ausgangsspannung. Handelsüblich sind die in Tabelle 7.1 aufgeführten Typen. Die angegebene maximale Eingangsspannung gilt für Betrieb unter Volllast.

Bemerkenswert ist bei dieser Spannungsreglerserie der eingebaute doppelte Überlastungsschutz. Die Bausteine reagieren in

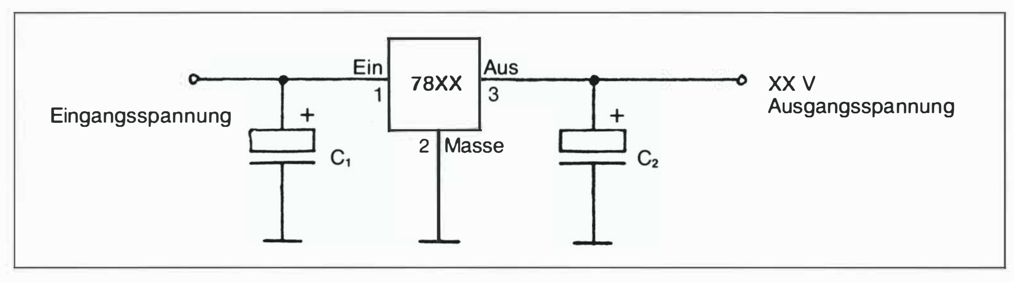


Bild 7.9: Standardschaltung mit einem Positivspannungsregler

Reglertyp	Ausgangsspannung	max. Eingangsspannung
7824	24 Volt	32 Volt
7815	15 Volt	24 Volt
7812	12 Volt	18 Volt
7808	8 Volt	12 Volt
7805	5 Volt	10 Volt

Tabelle 7.1: Gängige integrierte Festspannungsregler

zweifacher Weise auf Überlastung. Bei plötzlich auftretenden Überströmen mindert und begrenzt die Innenschaltung den Strom auf einen erträglichen Wert. Bei thermischer Überlastung jedoch schaltet sie – ähnlich wie ein Sicherungsautomat – ganz ab. Damit ist das Netzteil ohne weitere Maßnahmen bereits Kurzschlußfest.
Standardregler der 78XX-Serie vertragen einen Dauerstrom von 1A. Dazu müssen sie allerdings auf ein Kühlblech montiert werden. Neben ihnen sind auch Ausführungen für diverse andere Maximallasten lieferbar. In Kapitel 1.8.1 haben wir bereits ein Netzteil mit dem 2A-Typ 78S05 aufgebaut. Die Anschlußbelegung der Regler-ICs entnehmen Sie Anhang A.
Bild 1.29 zeigt das Experimentiernetzteil auf der Zählerplatine. Denkt man sich die vier Bauteile zur Frequenzaufbereitung im unteren Teil der Abbildung fort, erhält man die 78XX-Standardschaltung. Nach dem Trafo folgt ein Brückengleichrichter und sowohl Ein- als auch Ausgang des Reglers sind mit Kondensatoren abgeblockt.
Damit die Schaltung überhaupt richtig arbeiten kann, muß der vorgeschaltete Transformator neben der richtigen Spannung auch genügend Strom zur Verfügung stellen. Andernfalls würde die Spannung bei größerer Last einfach zusammenbrechen.
In gleicher Weise wie die positiven Festspannungsregler 78XX arbeiten die Negativregler 79XX, mit dem Unterschied, daß hier der positive Pol an Masse liegt. Bild 7.10 zeigt eine Netzteilschaltung mit zwei komplementären Spannungsreglern, die eine stabile positive und eine stabile negative Spannung gegen Masse liefert. Beachten Sie, daß die Elektrolytkondensatoren

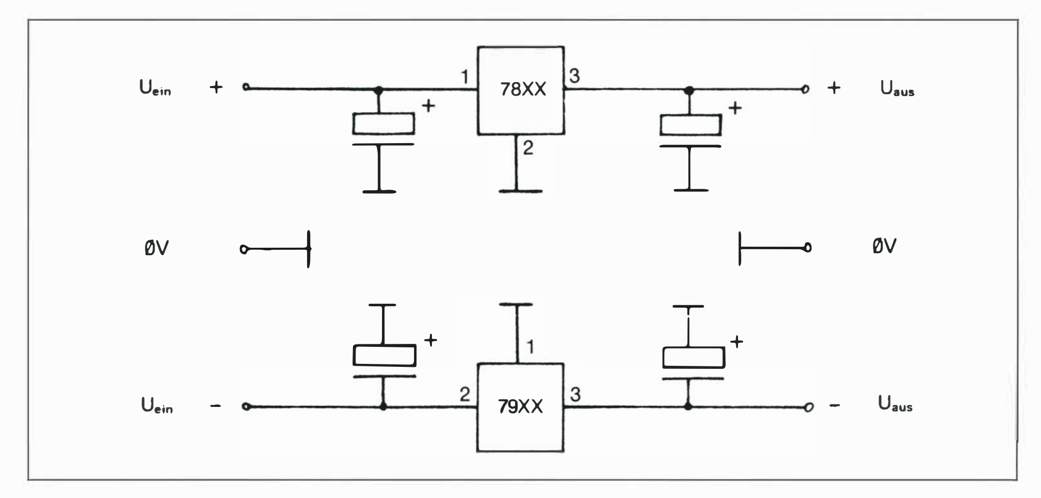


Bild 7.10: Symmetrische Spannungsversorgung

im negativen Zweig natürlich mit ihrem Pluspol an Masse liegen.
Das kurze BASIC-Programm NETZTEIL dient zum Berechnen der Bauteildimensionierung bei Standard-Netzteilschaltungen.

7.4 Regler überlistet

Bezugspotential jeder Schaltung ist Masse (0 Volt). Jeder Festspannungsregler hat einen Masseanschluß und stellt die Spannung am Ausgangspin so ein, daß zwischen diesen beiden Punkten genau der aufgedruckte Nennwert anliegt. Wird die Spannung am Massepin um einen genau definierten Wert gegen die tatsächliche Schaltungsmasse angehoben, dann erhöht sich auch die Reglerspannung um diesen Wert.
Im Verlauf dieses Buches wurden bereits mehrfach Zenerdioden eingesetzt. An ihnen liegt bei geeignetem Einsatz immer genau die jeweilige Zenerspannung. Setzt man nun eine solche Zenerdiode nach

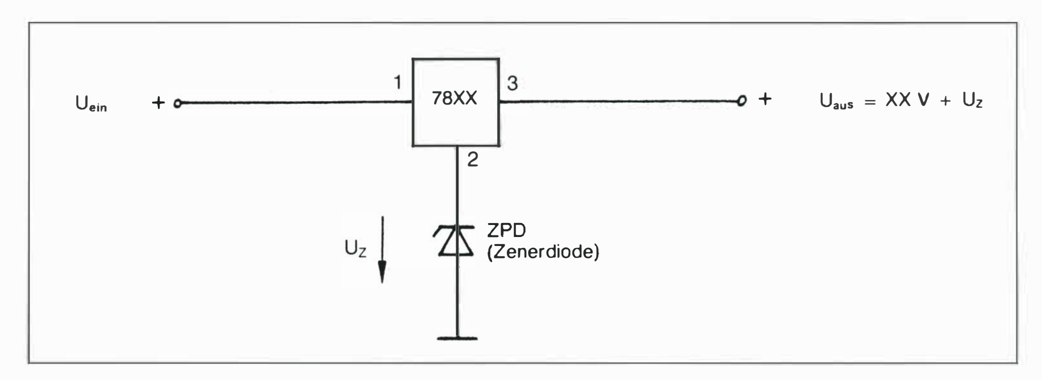


Bild 7.11: Erhöhung der stabilen Ausgangsspannung durch Zwischenschalten einer Zenerdiode

Bild 7.11 in den Massezweig des Reglers, dann erhöht sich die Ausgangsspannung um die Höhe der Zenerspannung.
Mit dieser Methode lassen sich praktisch alle geforderten Netzteilspannungen erzeugen.

7.5 Nutzung der Spannungen im C64

Der C64 selbst nutzt die Wechselspannung ebenfalls zur Erzeugung diverser Hilfsspannungen. In einem Zweig wird sie über eine Brückenschaltung gleichgerichtet und einem Spannungsregler (7805) zugeführt, dessen Ausgangsspannung die Bezeichnung +5V CAN trägt. Diese Spannung versorgt den Videocontroller, die Videoausgangsstufe und alle zur Takterzeugung benötigten ICs; kurz all das, was direkt mit relativ hohen Schaltfrequenzen zu tun hat. So ist eine wirkungsvolle Entkopplung der Betriebsspannungen gegeben.
Weiterhin wird aus der Wechselspannung über einen Einweggleichrichter eine unregulierte 9-V-Gleichspannung erzeugt, die zur Versorgung der Datasette benötigt wird.
Der Einweggleichrichter ist Teil einer Spannungsverdopplerschaltung, die über einen weiteren Regler (7812) eine konstante Spannung von 12 Volt liefert. Sie wird für den Videocontroller und die Audioausgangsstufe benötigt.
Außerdem besorgt eine kleine Wandlerschaltung die Umformung einer Wechselspannungshalbwelle in ein Rechtecksignal zur Ansteuerung der Echtzeituhren in den beiden CIA-Bausteinen 6526.
Alle anderen ICs wie auch die +5V-Anschlüsse an den Steckverbindern werden direkt von der bereits fix und fertig stabilisierten +5V-Spannung des Netzteils versorgt. Dieses Gerät ist für normalen Betrieb gerade noch ausreichend konzipiert. Man darf nicht von Strommessungen mit einem Vielfachgerät ausgehen, denn vor allem in der Digitaltechnik hat man es mit schnell wechselnden Lasten zu tun. Während innerer Schaltvorgänge wird mehr Strom verbraucht als im Ruhebetrieb. Dadurch entstehen Stromspitzen, die im Extremfall ein Vielfaches des Durchschnittsbedarfs betragen können. Ein Netzteil muß auch dann noch genügend Reserven mobilisieren können. Sind einige Erweiterungen angeschlossen, kann die Trafo- Spannung zeitweise bereits so stark absinken, daß Netzspannungsschwankungen – zum Beispiel beim Einschalten anderer Geräte – einen glatten Systemabsturz zur Folge haben. Um diesen Fall zu vermeiden, bleibt einem nichts anderes übrig, als dem Computer ein größeres Netzteil zu spendieren.

7.6 Mehr Leistung für den C64

Der Transformator ist das wichtigste und leider auch das teuerste Einzelteil eines Netzgerätes. Vor allem von ihm hängt der maximal entnehmbare Strom ab. Um den C64 so weiterbetreiben zu können, wie bisher, muß auch das neue Netzteil die gerade angesprochenen Spannungen liefern können. Wir verwenden daher sinnvollerweise einen Trafo, der ebenfalls 9-Volt-Wechselspannung liefern kann. Kleine Abweichungen sind nicht tragisch: 8 Volt sind genauso verwendbar wie 10 Volt. Diese Spannung kann direkt an die Wechsel-

spannungspins des Netzteilsteckers gelegt werden.
Um einen sicheren Betrieb zu gewährleisten, sollte die maximale Stromstärke etwa 4A betragen können.
Bild 7.12 zeigt, wie es weitergeht. Über einen Brückengleichrichter wird zunächst eine ungeregelte Gleichspannung erzeugt. Wie gesagt gibt es Ausführungen von Festspannungsreglern, die auch höhere Lastströme vertragen. Ihr Nachteil ist allerdings ein recht hoher Preis. Sinnvoller ist es da, einen zusätzlichen Lasttransistor einzusetzen, der den Hauptstrom übernimmt. In unserem Fall tut das der Komplementärtyp zum bekannten 2N3055, der MJ2955, der überall preisgünstig zu haben ist. Über den Widerstand erhält er zwischen Emitter und Basis eine Vorspannung, die optimales Regelverhalten ermöglicht. Wie gewohnt, glätten die Kondensatoren die Ausgangsspannung.
Für größere Ströme muß der Lasttransistor natürlich auf einen Kühlkörper montiert werden. Er entwickelt bei Vollast schon beträchtliche Verlustwärme.

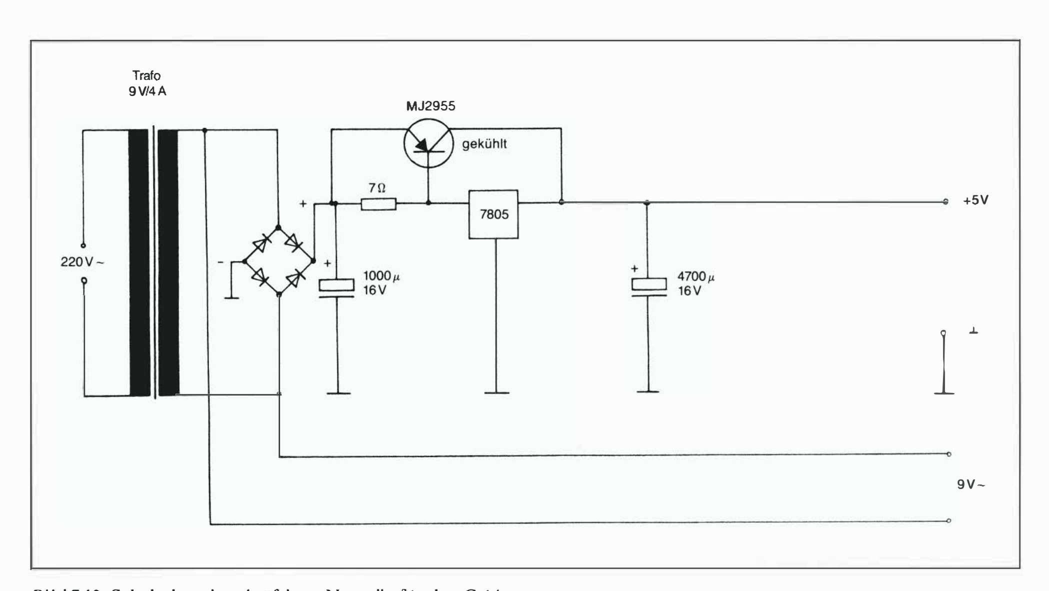


Bild 7.12: Schaltplan eines kräftigen Netzteils für den C 64

Kapitel 8 Die Control-Ports

Auf der rechten Gehäuseseite befinden sich neben dem Einschalter zwei gleichartige Anschlüsse: der Control-Port 1 und der Control-Port 2. Obwohl sie im wesentlichen gleich aufgebaut sind, gibt es doch Unterschiede im Detail, und mit dem Joystick sind die Einsatzmöglichkeiten noch längst nicht ausgeschöpft.

8.1 Einsatz von Joysticks

Das erste Zusatzgerät eines Heimcomputer-Besitzers ist meist ein Joystick. Glücklicherweise hat sich Commodore beim C64 für die allgemein übliche neunpolige Min-D-Steckverbindung entschieden, so daß gängige und dadurch preisgünstige Joystick-Ausführungen eingesetzt werden können, obwohl generell der ungeschützte Einbau der männlichen Steckerhälfte im Computer nicht die beste aller Lösungen darstellt. Die offenliegenden Pins können unbeabsichtigt mit dem Finger berührt werden, beispielsweise wenn man nach dem gleich nebenan befindlichen Schalter tastet. Ist man zufällig statisch aufgeladen, was bei synthetischer Kleidung und Teppichboden nicht selten vorkommt, kann das ganz schnell zur Zerstörung des angeschlossenen MOS-Bausteins CIA #1 führen!
Es ist empfehlenswert, die beiden Control-Ports mittels eines Pappstreifens abzudecken, der einseitig mit Tesafilm am Gehäuse befestigt ist, und bei Bedarf einfach hochgeklappt wird, falls nicht ohnehin ständig beide Buchsen beschaltet sind.
Die Funktion eines Joystick ist nicht schwer zu durchschauen. Er besitzt im Inneren fünf Taster (vier für den Steuerknüppel und einen für den Feuerknopf), die bei Betätigung den jeweiligen Steckeranschluß nach Masse kurzschließen. Die unterschiedlichen logischen Pegel können über Portanschlüsse des CIA # 1 abgefragt werden. Die genaue Pinbelegung der beiden Control-Ports können Sie Anhang C entnehmen. Wie aus den Tabellen hervorgeht und wie auch aus dem Schaltplan ersichtlich, sind die zuständigen Pins jedoch direkt parallel zur Tastatur verdrahtet. Wenn nicht besondere Vorkehrungen getroffen werden, kann also die Tastaturabfrage falsche Joystick-Werte hervorrufen. Will man sichere Aussagen erhalten, muß man während einer Joystick-Abfrage den Tastaturbetrieb abschalten. In einem BASIC-Programm kann das für Port 2 folgendermaßen aussehen:

```
10 POKE 56322,224: REM
TASTATUR ABSCHALTEN
20 X=PEEK(56320) : REM
JOYSTICK-ZUSTAND ERMITTELN
30 POKE 56322,255: REM
TASTATUR WIEDER
EINSCHALTEN
```

Für den Control-Port 1 ist grundsätzlich nur Zeile 20 zu ersetzen durch

```
20 X=PEEK(56321) .
```

Bevor Sie aber diese Zeilen in Ihr Programm einbauen, sollten Sie noch einen Blick auf Kapitel 8.3 werfen, in dem eine universelle Abfrageroutine in Maschinensprache vorgestellt wird.

8.2 Was es mit Paddles auf sich hat

Ein Joystick erlaubt nur relativ grobe Eingaben: links, rechts, oben, unten sowie Feuerknopf gedrückt oder nicht. Paddles dagegen sind Regler, die stufenlos verstellbar sind. Sie werden daher auch analoge Eingabegeräte genannt, während Joysticks immer digital arbeiten. Das Funktionsprinzip ist einfach und schnell erklärt: Der C64 besitzt zwei Analog/Digital-Wandler, die übrigens im Sound-Chip untergebracht sind. Ein solcher Wandler ermittelt zu einem Widerstandswert an seinem Eingang eine Bitkombination, die er jeweils in einem Register festhält.

Damit zwei Paddlepaare quasi gleichzeitig betrieben werden können, nämlich eines am Control-Port 1, das andere am Control-Port 2, verfügt der C64 über Analogschalter, mit denen die Poti-Anschlüsse des Sound-Chip umgeschaltet werden können. Bild 8.1 zeigt diese Anordnung im einzelnen. Die beiden Portleitungen PA6 und PA7 des CIA # 1 dienen zum Öffnen und Schließen jeweils zweier elektronischer Schalter im CMOS-IC 4066. Es ergibt sich das gleiche Problem wie schon bei der Joystick-Abfrage, denn auch diese beiden Anschlüsse werden bereits bei der Tastaturabfrage benutzt. Um alle Probleme der Doppelausnutzung dieser Ports endgültig zu beseitigen, soll im folgenden eine kurze Maschinenroutine vorgestellt werden, die auch von BASIC aus universell einsetzbar ist.

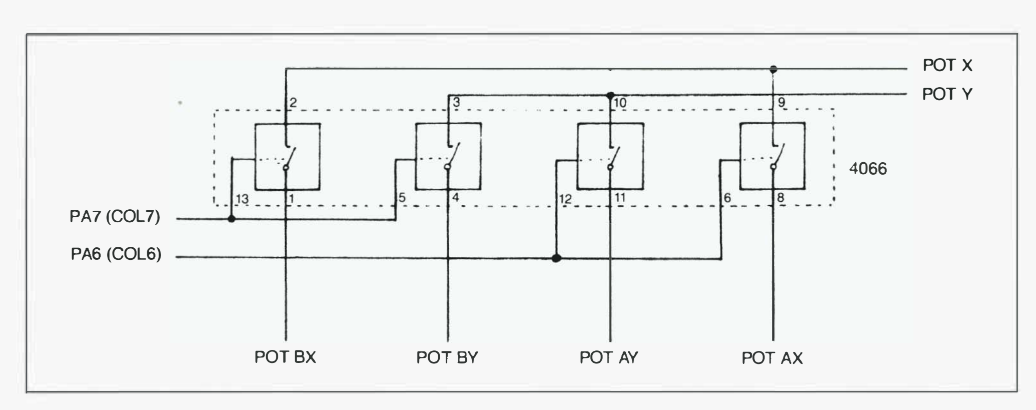


Bild 8.1: Die Analogschalter zur Auswahl des Paddlepaares

Adresse hex.	dez.	Inhalt
$CFFF	53247	Port 1, X-Wert
$CFFE	53246	Port 1, Y-Wert
$CFFD	53245	Joystick Port 1
$CFFC	53244	Port 2, X-Wert
$CFFB	53243	Port 2, Y-Wert
$CFFA	53242	Joystick Port 2

Tabelle 8.1: Ausgaberegister der Abfrageroutine

8.3 Eine universelle Joystick- und Paddle-Abfrage

Das Assemblerprogramm CONTROLPORT.ASM wurde für den Kassettenpuffer ab $033C (dezimal 828) geschrieben. Wird es aufgerufen, dann ermittelt es alle Paddle- und Joystick-Zustände und stellt sie in den sechs obersten Adressen des $C-Bereiches ($C000 bis $CFFF) zur Verfügung. Eine genaue Aufstellung enthält Tabelle 8.1.
Während des Maschinenprogramms wird mit CLI der Systeminterrupt (IRQ) und damit die Tastaturabfrage außer Gefecht gesetzt. Das funktioniert, weil die Ports nur während der Interrupt-Routine beeinflußt werden. Mit dem Unterprogramm LIES erfolgt die Umschaltung des jeweiligen Control-Port auf die Analogeingänge und nach Abwarten der Wandlungszeit die Übernahme der ermittelten Werte ins X- und Y-Register. Anschließend werden sie in die endgültigen Register geschrieben und noch jeweils das aktuelle Joystick-Byte geholt. Der Rücksprung erfolgt, nachdem alle Leitungen wieder in ihren ursprünglichen Zustand zurückversetzt wurden.

Die beschriebene Routine läßt sich auch von BASIC aus benutzen. Wie dabei vorzugehen ist, zeigt das Demoprogramm CONTROLPORT.BAS. Gleich nach dem Start sollte zunächst das Unterprogramm ab Zeile 10 000 aufgerufen werden. Es installiert die Abfrageroutine im Kassettenpuffer. Soll im Programmverlauf dann ein Joystick- oder Paddle-Wert eingelesen werden, ruft man mit

```
SYS 828
```

diese Maschinenroutine auf und kann auf die ermittelten Werte nach Tabelle 8.1 zugreifen. Bild 8.2 zeigt die entstehenden

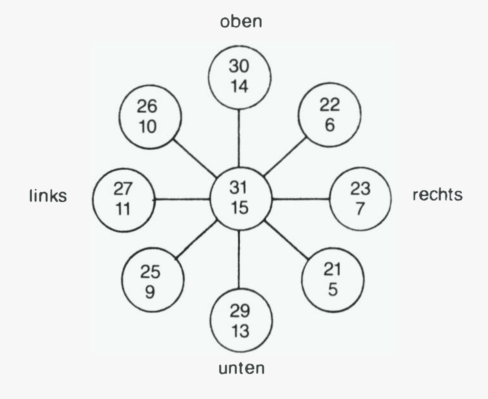


Bild 8.2: Abfragewerte bei den erreichbaren Joystick-Richtungen. Untere Zahlen bei gedrücktem Feuerknopf.

Werte für alle realisierbaren Joystick-Stellungen, oben normal und unten mit gleichzeitig gedrücktem Feuerknopf.

8.4 Paddles selbstgebaut

8.4.1 Fast rein mechanisch

Bereits ein regelbarer Widerstand, vom Analogeingang nach +5V geschaltet, sollte über den Wandler eigentlich genau einstellbare Abtastwerte liefern. Die Analogschalter im CMOS-IC 4066 wurden aber nicht speziell zur Übertragung von Widerstandswerten entwickelt, sondern eigentlich zum Schalten von Spannungen. Versuche ergaben, daß bei Potis der an sich offene Anschluß an Masse gelegt werden sollte. Dadurch liegt eine regelbare Spannung am Schleifer, die vom Analogschalter offensichtlich besser verarbeitet werden kann, als lediglich ein Widerstand nach +5V. Wie im Schaltplan (Bild 8.3) zu sehen ist, wurde im Paddle zusätzlich noch ein Trimmpotentiometer vorgesehen, das als Begrenzer wirkt und für die vollständige Ausnutzung des Regelbereiches

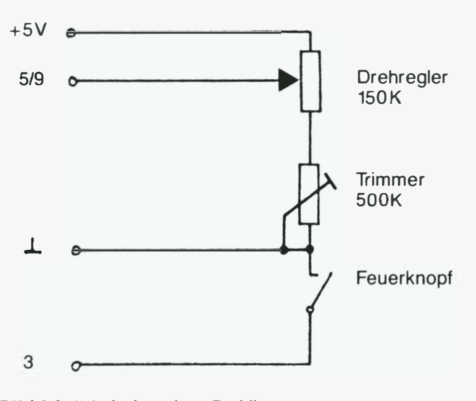


Bild 8.3: Schaltplan eines Paddles

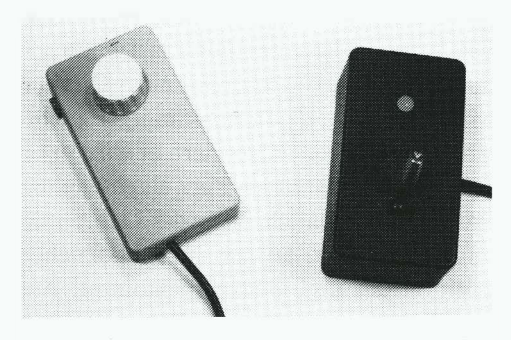
sorgen soll. Andernfalls wäre eventuell bereits bei halb geschlossenem Regler der Maximalwert 255 erreicht. Die Dimensionierung des Trimmpotentiometers richtet sich nach dem des Reglers selbst. Versuche ergaben für einen Reglerwert von 150 Kiloohm beispielsweise einen Trimmpotentiometer-Wert von 500 Kiloohm. Bei einem Regler mit 10 Kiloohm genügen 50 Kiloohm. Außerdem enthält die Schaltung einen Feuerknopf, der bei Paddles nicht der Verdrahtung in Joysticks entspricht.

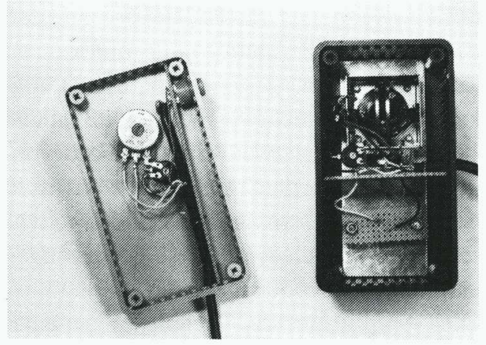
Zum Einbau des Reglers wird ein handliches und formschönes Gehäuse mit den ungefähren Abmessungen 160 x 60 x 20 mm ausgewählt, das an den Innenflächen der Seitenwände Stege zum Einstecken von Platinen aufweist.

Zuerst werden drei Bohrungen angebracht, und zwar für das Drehpotentiometer auf der Frontseite, für den Feuerknopf an der linken (bzw. für Linkshänder rechten) Seitenwand oben, und für die Durchführung der Zuleitung an der unteren Stirnseite.

Ein schmaler Streifen wird nun aus einer Lochrasterplatte gebrochen, der Feuerknopf darauf gesteckt und die Anordnung in das Gehäuse eingepaßt. Es ist dabei empfehlenswert, einen runden Knopf zu verwenden, um die benötigte Öffnung im Gehäuse leicht herstellen zu können. Auch sollte der Taster einen deutlichen Druckpunkt besitzen; etwa wie der Feuerknopf eines Joystick auch. Als Drehpotentiometer eignet sich am besten eine Ausführung mit Metallachse, die besonders angenehme Bedienungseigenschaften aufweist. Diese Achse wird auf die benötigte Länge (je nach Knopf) abgesägt und der Regler festgeschraubt.

Die abisolierte Zuleitung kann mittels zweier abisolierter Schaltdrähtchen an der Lochrasterplatte festgeklemmt werden. Auf den Fotos 8.1 und 8.2 kann man alle genannten Teile erkennen.





Fotos 8.1 und 8.2: Eindimensionale und zweidimensionale Paddles

Die Anordnung wird nach dem Schaltplan (Bild 8.3) verlötet, wobei das Trimmpotentiometer freitragend eingefügt werden kann. Wenn man will, kann man die Eingänge für X- und Y-Wert beide am Schleifer anlöten.

8.4.2 Abgleich für optimalen Arbeitsbereich

Bevor der Deckel aufgeschraubt wird, sollten Sie noch die Abgleicharbeiten durchführen. Dazu kann bereits das kurze Demoprogramm CONTROLPORT.BAS Verwendung finden. Laden Sie es in den Computer und stöpseln Sie den Regler in einen der beiden Ports.

In der Grundstellung sollte die Zahl für den Feuerknopf 31 sein; wird er gedrückt, dann muß dort eine 27 auftauchen.

Zum Abgleich wird der Drehregler ganz an den rechten Anschlag gedreht. Der Wert des Trimmpotentiometers im Gehäuse-Inneren sollte nun so verstellt werden, daß auf dem Bildschirm für den entsprechenden Wert gerade noch 255 angezeigt wird. Damit ist der Paddle-Zusatz fertiggestellt und nach Aufschrauben des Deckels einsatzbereit.

In der gleichen Weise kann auch ein X-Y-Paddle hergestellt werden, indem man jedes der beiden Potis eines Kreuzknüppels nach Bild 8.3 verschaltet, und den einen Ausgang an Pin 5 des Control-Port-Steckkers, den anderen an Pin 9 anschließt.

8.5 Extrawurst – der Control-Port 1 und der Video-Chip

8.5.1 Nicht nur für den Lightpen

Wenn Sie die Belegungspläne der beiden Control-Ports genau vergleichen, wird Ihnen sicherlich ein wesentlicher Unterschied auffallen. Der Eingang für den Feuerknopf bei Port 1 ist zusätzlich noch mit dem Lightpen-Eingang des Videocontrollers verbunden. Unter Lightpen (= Lichtgriffel) versteht man einen handlichen Stift, der zur Eingabe oder Bestimmung eines Punktes auf dem Bildschirm dient, ohne daß umständlich auf der Tastatur gehämmert werden muß. Es ist möglich, durch simples Auflegen der Stift-

spitze auf die Mattscheibe des Fernsehers oder Monitors, Kontakt zum Programm herzustellen.
Im Kapitel 5.1 wurde schon besprochen, wie das Bild auf einem Fernseher oder Monitor entsteht. Dabei läuft ein Lichtpunkt Zeile für Zeile über den Schirm und wird vom Computer in seiner Helligkeit verändert. Natürlich weiß der Videocontroller zu jeder Zeit, welchen Punkt des Bildspeichers er gerade ausgeben muß. Hält man nun den Lightpen an eine beliebige Position des Bildes, erzeugt der Lichtpunkt beim Vorbeilaufen an dieser bestimmten Stelle einen kurzen Impuls, und der Controller übernimmt daraufhin die augenblicklichen Bildspeicherkoordinaten in zwei Register, die man anschließend bequem auslesen kann. Zuständig sind die beiden Speicherzellen $D013 (53 267) und $D014 (53 268).
Ähnlich wie bei bestimmten Ereignissen in den CIA-Chips können Sie auch bei Lightpen-Impulsen über den Videocontroller einen maskierbaren Interrupt (IRQ) auslösen. Dazu muß im Register $D01A (53 274) das Bit 3 gesetzt sein, um die Interruptquelle freizugeben. Ein Impuls vom Lightpen setzt Bit 3 im Register $D019 (53 273).
Gerade diese Interruptmöglichkeit kann unabhängig von der Video-Funktion auch bei anderen Zusätzen am Control-Port 1 ausgenutzt werden. So kann man zum Beispiel bereits über den Feuerknopf eines angeschlossenen Joystick einen IRQ auslösen, was sicherlich nicht nur für Spiele interessant ist.

8.5.2 Fang den Punkt!

Was Impulse am Lightpen-Eingang bewirken, wissen Sie jetzt, doch wie bekommt man aus einem wandernden Lichtpunkt einen Impuls?
Die Antwort ist eigentlich klar: natürlich durch eine Art Lichtschranke. Diesmal wird allerdings nicht ein ständiger Lichtstrahl unterbrochen, sondern er wird in regelmäßigen Abständen kurz eingeschaltet, immer dann nämlich, wenn der Lichtpunkt auf dem Schirm an der augenblicklichen Position des Lightpen vorbeikommt. Aus den Bemerkungen über die Entstehung eines Fernsehbildes aus Kapitel 5 folgt, daß der entstehende Lichtimpuls nur extrem kurz sein kann. Zudem gibt die Helligkeitsänderung nicht viel her, so daß an den Lichtaufnehmer zwei wichtige Forderungen gestellt werden müssen: erstens muß er sehr schnell ansprechen und zweitens sehr empfindlich sein. Diese Forderungen erfüllen nur Fototransistoren.
Die Amplitude der Schaltflanke eines Fototransistors allein reicht bei weitem nicht aus, um den Lightpen-Eingang unseres Videocontroller ansteuern zu können. Daher wird ein empfindlicher Verstärker zwischengeschaltet, an dessen Ausgang TTL-Pegel entsteht. Prinzipiell wäre mit dieser Anordnung bereits Lightpen-Betrieb denkbar. Man muß sich jedoch klarmachen, daß die Zeilen eines Fernsehbildes sehr dicht untereinander geschrieben werden. Aufgrund von Streuungen und Reflexen an der Mattscheibe ist es unmöglich, mit dem Fototransistor nur genau eine einzige Zeile zu erfassen. Man bekäme einen Impuls von der ersten Zeile, 64 Mikrosekunden später erneut einen

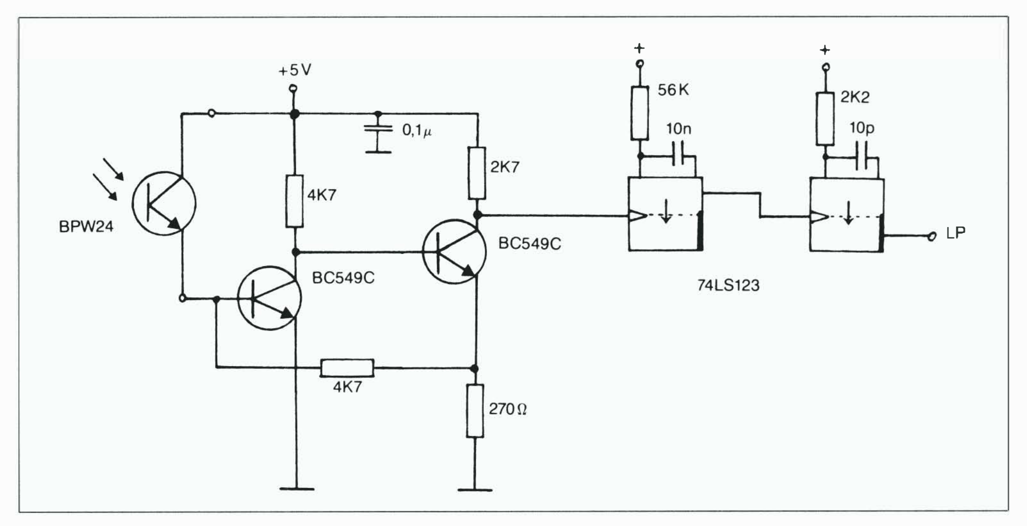


Bild 8.4: Schaltung des Lightpen

Impuls von der Zeile darunter und eventuell sogar noch einen dritten.
In unserer Schaltung nach Bild 8.4 werden daher zwei Monoflops eingesetzt, die nur beim ersten Ansprechen einen Impuls an den Computer abgeben, indem das erste Monoflop etwa für die nächsten vier Zeilen im HIGH-Zustand bleibt. Das zweite Monoflop erzeugt aus der positiven Schaltflanke einmalig einen kurzen Impuls, der an den Computer weitergeleitet wird.

8.5.3 Lightpen selbstgebaut

Die Lightpen-Schaltung nach Bild 8.4 ist leicht aufzubauen. Bild 8.5 zeigt das Platinenlayout und Bild 8.6 den zugehörigen Bestückungsplan. Richten Sie sich bitte nach den Hinweisen im Anhang A. Die benötigten Bauteile sind in der Bestückungsliste (Tabelle 8.2) zusammengestellt. Es ist unbedingt empfehlenswert, den angegebenen Transistortyp zu benutzen. Das nachgestellte C klassifiziert den Verstärkungsfaktor. Auch beim angegebenen Fototransistor handelt es sich um eine Ausführung mit speziellen Eigenschaften in bezug auf Ansprechempfindlichkeit und Reaktionszeit.

1	Fototransistor BPW 24
2	Transistoren BC549C
1	Doppelmonoflop 74LS123
1	Sockel 16polig
1	Widerstand 56 Kiloohm
2	Widerstände 4,7 Kiloohm
1	Widerstand 2,7 Kiloohm
1	Widerstand 270 Ohm
1	Kondensator 0,1 Mikrofarad
1	Kondensator 10 Nanofarad
1	Kondensator 10 Pikofarad
1	Min-D-Stecker, 9polig, female mit abgewinkelten Kontakten
1	Platine nach Bild 8.6

Tabelle 8.2: Bauteile für den Lightpen

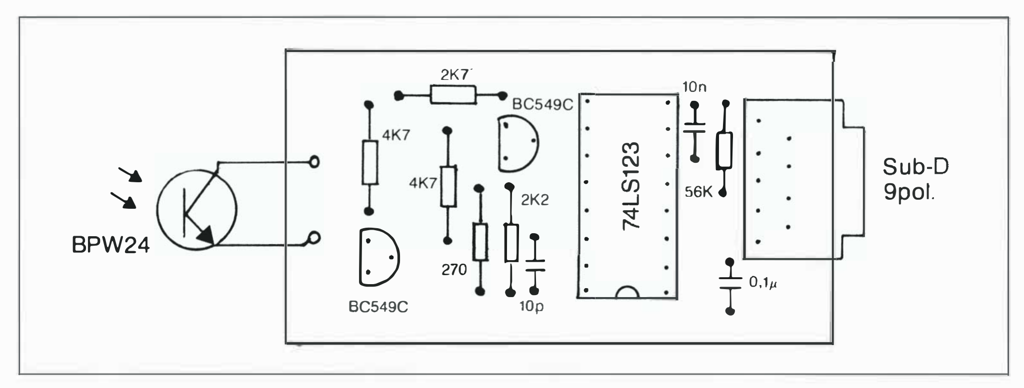


Bild 8.5: Bestückungsplan des Lightpen

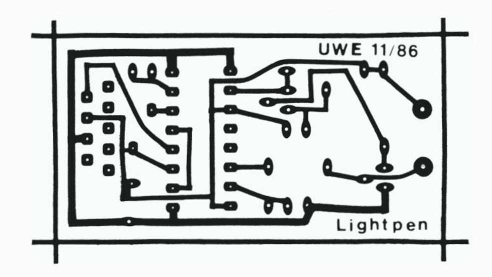


Bild 8.6: Die Lightpen-Platine

Der Min-D-Stecker kann direkt auf die Platine gelötet werden. Die ganze Anordnung ist dann unmittelbar an den Control-Port 1 ansteckbar. Natürlich läßt sich auch ein ähnlicher Stecker über Kabel verwenden.

Es ist sinnvoll, den Fototransistor in einen alten Kugelschreiber einzubauen. Dazu wird die Mine entfernt und das Loch in der Spitze so weit aufgebohrt, bis sich der Fototransistor von innen einpressen läßt. Der Emitteranschluß ist am Gehäuse durch eine kleine Metallfahne gekennzeichnet.

Das kurze BASIC-Programm LIGHT-PEN-TEST ermöglicht es, die Funktion des Zusatzes gleich auszutesten. Nach Starten mit RUN erscheint die augenblickliche Lightpen-Position oben links auf dem Bildschirm. Bewegt man den Fototransistor vor dem Bildschirm, müssen sich die angezeigten Zahlen entsprechend verändern.

8.5.4 Tricks gegen den springenden Punkt

Besonders bei älteren Fernsehgeräten kann es vorkommen, daß die ermittelte Position des Lightpen trotz der beschriebenen schaltungstechnischen Vorkehrungen stark schwankt. Das könnte eine Auswirkung der elektrischen und magnetischen Felder im Inneren der Röhre sein. Sie erzeugen im Fototransistor auch dann Impulse, wenn gar kein Lichtpunkt in Sicht ist. Um diesen Effekt zu vermeiden, sind zwei Dinge möglich.

Zunächst läßt sich der Fototransistor abschirmen, indem er in irgendeiner Form mit Metall ummantelt wird, das elektrisch mit der Computermasse verbunden ist. Meist wird der vorgeschlagene Fototransistor ohnehin in einem Metallgehäuse geliefert. Lötversuche an diesem Gehäuse sollten unterbleiben, denn sie könnten den

Halbleiter schnell überhitzen und damit ganz unbrauchbar machen. Besser ist das Einpressen des Bauelements in eine leitende Aufnahme.

Die andere Möglichkeit geht von der Idee aus, den optoelektronischen Sensor räumlich möglichst weit vom Bildschirm zu trennen. Trotzdem müssen nach wie vor die Lichtsignale abgetastet werden.

Die Lösung dieses scheinbaren Widerspruchs heißt Lichtwellenleiter. Kurze Abfallstücke aus der industriellen Herstellung von optischen Kabeln sind inzwischen beim Elektronik-Versandhandel sehr günstig zu haben. Man kann sie zum Schutz vor Knicken in eine Kabelisolierung stecken.

Noch eine andere Methode kann den springenden Punkt etwas zur Ruhe bringen. Es ist klar, daß durch Unlinearitäten beim Bildaufbau und durch leichtes Kippen des Lightpen während der Abfrage nicht bei jedem Einzelbild die gleichen Koordinaten ermittelt werden. Das Fernsehbild wird hierzulande alle 20 Millisekunden neu aufgebaut. Genauso oft entsteht auch ein neuer Abtastwert. Für eine saubere Auswertung – beispielsweise in einem Malprogramm – genügen aber weit weniger Abtastwerte. Man sollte in einem solchen Fall immer mehrere Abtastungen mitteln, bevor man einen Punkt ausgibt. Maschinenroutinen sind dafür allemal schnell genug.

Anhang A
Platinenherstellung und Bestückung

A.1 Praktische Tips zum Aufbau der Platinen

Bevor Sie mit dem Nachbau der beschriebenen Schaltungen beginnen, sollten Sie diesen Anhang unbedingt aufmerksam durchgelesen haben. Er enthält Grundlagen zum Umgang mit elektronischen Elementen, wertvolle Tips für die Platinenherstellung, eine Einführung ins Löten sowie allgemeine Beschreibungen der wichtigsten verwendeten Bauteile samt ihrer Pinbelegung.

A.1.1 Der Arbeitsplatz

Zunächst brauchen Sie einen Tisch mit genügend Platz für die Materialien und zum Arbeiten, den Sie auch mal unaufgeräumt stehenlassen können, eine Steckdose in der Nähe und ausreichend Licht, weil Sie es mit vielen kleinen Bauteilen zu tun bekommen werden. Als Arbeitsunterlage kann eine alte Schreibtischmatte dienen, damit nichts Wertvolles durch herabfallende Lötzinntropfen angeschmort wird. Natürlich sollte auch der Computer nicht allzuweit weg stehen.
An dieser Stelle noch ein dringender Hinweis:
Die meisten Schaltungen in diesem Buch arbeiten mit so kleinen Spannungen, daß sie völlig ungefährlich sind. Die wenigen Ausnahmen sind entsprechend gekennzeichnet. Achten Sie in solchen Fällen besonders sorgfältig darauf, daß niemand eine offenliegende und von Ihnen unbewachte Schaltung berühren kann. Trennen Sie die Schaltung vorsichtshalber auch dann vom Netz, wenn Sie den Elektronik-Arbeitsplatz nur für wenige Augenblicke verlassen, oder noch besser, stecken Sie den Netzstecker nur in die Dose, wenn die entsprechende Schaltung ganz in ein sicher isolierendes Gehäuse eingebaut ist.

A.1.2 Das richtige Werkzeug

Elektronikarbeiten – gerade bei Zubehör für Computer – sind Feinarbeiten. In diesem Buch werden vorwiegend Platinen vorgestellt, die gar nicht erst in Gehäuse eingebaut werden sollen, da sie an der Rückseite des Computers angesteckt werden, und dort auch so sicher genug sind. Wir können uns also im großen und ganzen auf das reine Elektronikwerkzeug beschränken. Ein wunder Punkt ist allenfalls die Platinenherstellung, auf die wir gleich noch genauer zu sprechen kommen.
Wichtig ist eine Elektronikflachzange zum

Abwinkeln der Anschlußdrähtchen vor dem Bestücken. Ein kleiner Saitenschneider, um die überstehenden Anschlußdrähtchen der Bauelemente abzuzwicken, nachdem sie in die Platine gesteckt wurden. Ein scharfes Bastelmesser und eine Abisolierzange helfen beispielsweise beim Vorbereiten von Litzen. Weiterhin sollte noch ein Schraubenzieher und eine Pinzette zur Verfügung stehen.
Wichtigstes Utensil ist natürlich der Lötkolben. Wie bereits im Vorwort angedeutet, dürfen nur Feinlötkolben mit dünner, verzinnter Spitze eingesetzt werden. Ihre Leistung beträgt etwa 15 bis 30 Watt. Zu hohe Temperaturen können die empfindlichen Halbleiterbauelemente leicht zerstören.
Fast alle hochintegrierten ICs sind heute auf MOS-Basis gefertigt (Metall Oxide Semiconductors = Metalloxid-Halbleiter). Diese Bausteine nehmen Spannungsspitzen sehr ernst. Es genügt bereits eine kurze Schaltspitze auf der Netzleitung während des Lötvorgangs, um ein solches IC zu zerstören. Aus diesem Grund sollte man sich besser einen Lötkolben mit Trenntrafo – auch Lötstation genannt – zulegen. Dort werden die gefährlichen Spitzen heruntertransformiert und können keinen Schaden mehr anrichten.
Neben dem eigentlichen Werkzeug braucht man noch einige Meßinstrumente, um die richtige Funktion eines Schaltungsteils überprüfen zu können. Hier sollten Sie mindestens über ein Vielfachmeßgerät verfügen. Es sollte einen Ohm-Meßbereich haben, um die Leiterplatten auf Lötbrücken bzw. Unterbrechungen kontrollieren zu können, sowie mindestens einen Spannungsmeßbereich und einen Strommeßbereich.

A.2 Platinen selbstgemacht

A.2.1 Vom Layout zur gedruckten Schaltung

In diesem Buch befinden sich Vorlagen zur Herstellung gedruckter Schaltungen. Sie haben oft dünne und dicht nebeneinanderliegende Bahnen und erfordern daher sorgfältige Behandlung. Sollten Sie sich die Herstellung von Platinen nicht zutrauen, können Sie fertige Leiterplatten von der Firma Wempe Elektronic, Bahnhofstraße 5, 6430 Bad Hersfeld, bestellen. Fertige Bausätze sind über die Firma Schmotz Elektronik, Postfach 1412, 8202 Bad Aibling zu beziehen. Lesen Sie jedoch zuvor die folgende Anleitung. Sie werden sehen: So schwierig ist die Sache eigentlich gar nicht.
Grundmaterial zur Herstellung von gedruckten Schaltungen ist immer ein isolierender Träger (Leiterplatte) der – je nach Anwendung – einseitig oder beidseitig mit Kupfer beschichtet ist. Zusätzlich ist eine Positiv-Fotolack-Schicht aufgebracht, die das Kupfer abdeckt.
Dort, wo diese Schicht belichtet wird, löst sie sich beim Entwickeln ab und gibt die Kupferschicht frei. Beim anschließenden Ätzen wird das Kupfer an den ungeschützten Stellen vom Träger abgelöst und übrig bleiben nur die Leiterbahnen.

A.2.2 Bilder aus Kupfer

Erste Hürde bei der Platinenherstellung ist die richtige Belichtung. Die Platten können bedenkenlos bei gedämpftem Tageslicht verarbeitet werden. Nehmen Sie ein genügend großes Stück fotopositiv beschichtetes Basismaterial, ziehen Sie die schwarze

Schutzfolie ab und legen Sie den gewünschten Ausschnitt der Layout-Folie darauf. Um sicherzustellen, daß die Folie wirklich plan aufliegt, wird sie mit einer sauberen Glasplatte beschwert. Das ganze beleuchten Sie etwa 6 bis 8 Minuten lang mit einer 500-Watt-Halogen-Kopierlampe, die im Abstand von etwa 25 cm genau darüber angebracht ist. Leichte Überbelichtung ist besser als Unterbelichtung, denn dabei wird später das entwickelte Bild nicht schleierfrei und sauber oder die Schicht schwimmt fort.

Viele Hobby-Elektroniker sind bei doppelseitigen Platinen skeptisch. Das ist jedoch gar nicht nötig, wenn man einige Tricks beachtet. Zuerst wird an zwei angrenzende Ränder der Layout-Unterseite mit Tesafilm genau rechtwinklig je ein schmaler Abfallstreifen Basismaterial geklebt. Nun legt man den Oberteil des Layout von der Gegenseite her auf die Streifen, hält das ganze gegen das Licht und bringt die zueinander gehörenden Lötaugen zur Deckung. Genau in dieser Lage wird nun auch die Oberseite des Layout an die Streifen geklebt. So erhält man eine Tasche, in die das zu belichtende Material eingesteckt werden kann. Es sollte etwas größer sein als die Layouts, damit man es auf jeder Seite ebenfalls mit einem Streifen Tesafilm gegen Verrutschen sichern kann. Ober- und Unterseite werden jetzt nacheinander wie einseitige Platinen belichtet. Beim Herausnehmen bleibt die Tasche erhalten, so daß bei der nächsten Platine das erneute Justieren entfällt.

Sie werden sicherlich feststellen, daß die Lampe bei der Belichtung enorme Hitze entwickelt. Auch die der Lampe zugewandte Platinenseite erwärmt sich. So kommt es, daß bei doppelseitigen Platinen die zuletzt belichtete Seite wesentlich wärmer ist, als die andere. Würde man das Material sofort entwickeln, geschähe das in unterschiedlicher Geschwindigkeit, und die Aktion wäre zum Scheitern verurteilt.

Daher legen Sie die belichtete Platine zum Temperaturausgleich zunächst einmal mehrere Minuten bei gedämpftem Tageslicht oder bei Dunkelkammer-Rotlicht in Wasser.

A.2.3 Entwicklungshilfe

Nun kommt der kritischste Vorgang: das Entwickeln. Dazu wird die Platine in ein Natriumhydroxid-Bad getaucht, das auch als Ätznatron bekannt ist. Natriumhydroxid wird in Perlform geliefert und ist leicht in Wasser löslich, doch Achtung! Beim Lösen entstehen unangenehme Dämpfe, und die Flüssigkeit wird erstaunlich warm. Vermeiden Sie unbedingt Berührungen mit dieser Chemikalie, ziehen Sie einen alten Kittel an und bedecken Sie den Arbeitstisch mit einer dicken Lage Zeitungen. Die genaue Dosierung beim Lösen ist auf der Flasche angegeben und sollte unbedingt eingehalten werden.

Je nach Zustand der Lösung dauert der Entwicklungsvorgang meist nur wenige Sekunden. Sein Ende erkennt man daran, daß an den belichteten Stellen das blanke Kupfer zu sehen ist. Man darf die Platine jedoch nicht zu lange entwickeln, sonst wird der Lack auch dort abgelöst, wo er eigentlich stehenbleiben müßte. Zur Kontrolle nehmen Sie die Platine immer wieder aus der Flüssigkeit und spülen sie unter klarem Wasser ab. Wichtig ist, daß die Löcher in den Lötaugen zu erkennen sind, denn sie

müssen später unbedingt freigeätzt werden. Fassen Sie jetzt nicht mehr auf die entwickelten Bahnen, sonst kann es passieren, daß an einigen Stellen das Kupfer durch anhaftendes Fett nicht sauber weggeätzt wird.
Dauert die Entwicklung zu lange, ist die Platine entweder unterbelichtet, oder die Entwicklerlösung zu schwach oder verbraucht, geht sie zu schnell, ist dagegen die Lösung zu stark (mit Wasser verdünnen) oder das Bad zu warm (über 30 Grad C).

A.2.4 Jetzt wird's ätzend

Inzwischen können Sie die Ätzlösung ansetzen. Das geht mit Eisen-III-Chlorid oder mit Ätzsulfat. Beide Mittel werden als Granulat geliefert und erzeugen ihre volle Ätzkraft bei Temperaturen zwischen 45 und 50 Grad C. Eisen-III-Chlorid besitzt dabei den Nachteil schnell nachlassender Ätzkraft, rotbrauner Färbung (schlechter Sichtkontakt) sowie der Entwicklung von unangenehmen Gerüchen und Dämpfen.
Ätzsulfat dagegen ergibt eine klare Lösung, die nicht riecht und keine giftigen Dämpfe erzeugt. Auch hier sollte man Kontakt mit Haut und Augen sowie Textilien vermeiden, gegebenenfalls sofort mit lauwarmem Wasser und Seife abspülen.
Man kann das Granulat einfach in heißem Wasser lösen. Dann muß es jedoch vor dem Abkühlen verarbeitet werden. Profis verwenden Ätzanlagen mit Heizung, mit denen die Arbeit natürlich weit weniger aufwendig ist.
Sorgen Sie dafür, daß sich die Flüssigkeit ständig bewegt. Nach etwa 5 bis 10 Minuten ist der Ätzvorgang beendet. Das vorher blanke Kupfer muß restlos von der Trägerschicht abgelöst sein. Wichtig ist vor allem, daß keine Verbindungen zwischen einzelnen Leiterbahnen mehr stehenbleiben. Achten Sie auch darauf, daß die Mittelpunkte der Lötaugen deutlich weggeätzt sind. Sie dienen nämlich beim anschließenden Bohren als Zentrierung.
Ändert sich am Zustand nichts mehr, dann nehmen Sie die Platine aus dem Bad, spülen sie unter fließendem Wasser ab und trocknen sie mit einem saugfähigen Papier. Den restlichen Fotolack auf den Leiterbahnen sollte man zum Schutz vor Oxidation stehenlassen. Beim Löten brennt er sich an den erhitzten Stellen leicht fort und stört nicht.
Die benötigten Chemikalien können Sie mehrmals verwenden, bis die jeweilige Lösung verbraucht ist, was sich durch deutlich langsamere Reaktionen bemerkbar macht. Verbrauchte Ätzlösungen dürfen erst nach geeigneter Entgiftung ins Abwasser gelangen. Hierzu wird die Ätzlösung auf etwa ihr achtfaches Volumen mit Wasser verdünnt und unter ständigem Rühren zehnprozentige Natronlauge (pro Liter Ätzlösung zirka ein Liter Natronlauge erforderlich) zudosiert, bis sich ein pH-Wert von 10 in der Lösung einstellt. Das kann man beispielsweise mit einem Indikatorstäbchen überprüfen. Es entsteht ein voluminöser Niederschlag (Metallhydroxide), der sich langsam absetzt. Nach entsprechender Absetzzeit kann die entgiftete Lösung abgefiltert werden.

A.2.5 Bohren und Bestücken

Nun müssen noch die Löcher für die Bauteile gebohrt werden. Dazu braucht man auf jeden Fall einen Bohrständer, da die

feinen Bohrer sehr leicht abbrechen. Standard-Lochdurchmesser ist 0,8mm. Für größere Bauteile muß jedoch mit 1mm bzw. 1,3mm gebohrt werden. Diese Maße sollten möglichst eingehalten werden, damit sowohl elektrisch als auch mechanisch sichere Kontakte entstehen.

Nach dem Bohren wird eine Sichtkontrolle vorgenommen, indem man die Platine auf feine Leiterbahnunterbrechungen und Überbrückungen hin durchsieht. Unterbrechungen können mit einem kleinen Drahtstück oder einem Klecks Lötzinn geflickt und Kupferbrücken mit einem schmalen Schraubenzieher oder einer Reißnadel entfernt werden.

Auch beim Bestücken gibt es einige Grundregeln zu beachten. Da ist zum einen die Reihenfolge. In der Regel beginnt man mit den niedrigsten Elementen, also mit Drahtbrücken, falls diese vorhanden sind, fährt dann mit den Widerständen fort und gelangt danach zu IC-Sockeln, Kondensatoren und so fort. Zum Schluß kommen die empfindlichen Halbleiter an die Reihe, wie Transistoren und ICs. Man sollte sich angewöhnen, alle ICs zu sockeln. Das kostet vielleicht ein paar Mark mehr, aber man tut sich um vieles leichter, wenn einmal ein Baustein ausgewechselt werden muß.

Achten Sie besonders auf die korrekte Einbaurichtung von Halbleitern und Elkos. Einige Seiten weiter finden Sie Anschlußbelegungen vieler Bauteile angegeben. Sollten Sie abweichende Gehäuseausführungen erwischt haben, dann fragen Sie unbedingt beim Händler nach den entsprechenden Daten. Falsch eingebaute Teile können nicht nur eine ordnungsgemäße Funktion verhindern, sondern führen oft unversehens auch zur Zerstörung anderer Bauelemente.

A.3 Die Kunst des Lötens

Der Lötdraht des Elektronikers hat mit dem Lötdraht des Bauklempners nur wenig gemeinsam. Der glänzende Zinnmantel enthält eine gelblich-braune Masse: die Kolophoniumeinlage. Auf diese Masse kommt es beim Löten besonders an, denn sie ist das Flußmittel, das während des Erhitzens die Lötstelle von Oxidschichten und Verunreinigungen säubern soll, damit ein mechanisch fester und gut leitender Kontakt entsteht.

Der Lötkolben muß vor Beginn der Arbeiten Betriebstemperatur haben. Man prüft das, indem etwas Lötzinn auf die Spitze gehalten wird. Das Flußmittel sollte dabei sofort verdampfen und etwas Zinn auf der Spitze zerfließen.

Jedes Bauelement wird einzeln vorbereitet. Bei Widerständen geschieht das beispielsweise, indem die beiden Anschlußdrähtchen im richtigen Abstand rechtwinklig umgebogen werden. Man steckt anschließend die langen Drahtenden bis zum Anschlag durch die vorgesehenen Bohrungen und knickt sie von der Unterseite her leicht nach außen, damit das Element nicht wieder herausfallen kann. Beim Einlöten können Sie ja nichts mehr festhalten.

Was jetzt folgt, ist eigentlich gar nicht schwer. Es hört sich nur kompliziert an. Wichtig ist in jedem Fall Erfahrung, die erst nach einigen Versuchen kommen kann. Verzweifeln Sie also nicht gleich.

Falls irgendwie möglich, lagern Sie die Lötstelle so, daß die Schwerkraft den Lötvorgang unterstützt. Es ist günstig, wenn das

Zinn zur Lötstelle hinfließt und nicht von ihr weg. Sobald die Lötstelle ausreichend erwärmt ist, bringen Sie das Zinn an die Lötstelle heran. Dabei fließt zuerst das im Lötdraht eingebettete Kolophonium und dann erst das Zinn über die Lötstelle. Diese Reihenfolge ist vorgesehen.

Sobald genügend Zinn vom Lötdraht abgeschmolzen ist, nehmen Sie den Lötdraht von der Lötstelle weg. Wichtig ist, daß die Lötkolbenspitze gerade so lange an der Lötstelle gehalten wird, bis alle an der Lötung beteiligten Teile vom Zinn umflossen sind.

Wie bereits erwähnt, dürfen insbesondere Halbleiterbauelemente – also Transistoren, Dioden usw. – beim Einlöten nicht zu heiß werden. Die Lötzeit sollte daher möglichst kurz bleiben (maximal 5 Sekunden).

Die Lötstelle darf während des Erkaltens nicht mehr bewegt werden. Den Abkühlvorgang kann man beschleunigen, indem man den frischen Kontakt leicht anbläst. Eine fachgerechte Lötstelle glänzt silbrig und hat eine glatte Oberfläche.

Sollte beim Lötvorgang das Lot gut schmelzen, aber nicht ausreichend breit verlaufen, dann ist nicht genügend Flußmittel an die Lötstelle gelangt. Führen Sie noch etwas Lötdraht zu. Dasselbe tun Sie auch, wenn Sie eine erkaltete Lötstelle noch einmal bearbeiten wollen. Achten Sie aber darauf, daß nicht zu viel Lötzinn auf den Kontakt gelangt und eine unbeabsichtigte Lötbrücke zu anderen Leiterbahnen bildet. Zu viel Lötzinn und im Falle eines Falles auch eine Brücke kann mit einem Entlötgerät leicht abgesaugt werden.

Ist alles erfolgreich »verlaufen«, wird das überstehende Drahtende mit einem Saitenschneider abgeknipst.

So muß man nun nacheinander bei jeder Lötstelle vorgehen.

Wenn alles bestückt ist, sollte noch einmal eine Sichtkontrolle durchgeführt werden, denn es kann vorkommen, daß sich haarfeine Lötbrücken gebildet haben. Das geschieht leicht bei eng aneinander liegenden Lötstellen oder bei ICs mit Leiterbahndurchführungen zwischen den Anschlußbeinchen. Eine sorgfältige Kontrolle vor dem ersten Einschalten kann eine spätere, langwierige Fehlersuche ersparen, da Halbleiter leicht durch falsches Anschließen zerstört werden können, und dann unter Umständen gar nichts mehr geht.

A.4 Der unvermeidbare Kleinkram

Alle in den vorgestellten Schaltungen benötigten Bauteile kann man in Elektronikläden kaufen. Bei seltenen oder schwer beschaffbaren Teilen ist in der Bestückungsliste eine Bezugsadresse angegeben. Beachten Sie auch die Adressen von Versandgeschäften im Anhang D. Nicht immer werden Sie im Geschäft um die Ecke genau das Teil erhalten, das in der Liste aufgeführt ist. Wollen Sie zum Beispiel einen ganz bestimmten Transistor kaufen, kann es durchaus vorkommen, daß Sie gefragt werden, ob es auch ein Universaltyp sein kann, oder wenn Sie einen Widerstand mit 340 Ohm verlangen, kann es heißen: »Können es auch 330 Ohm sein?«

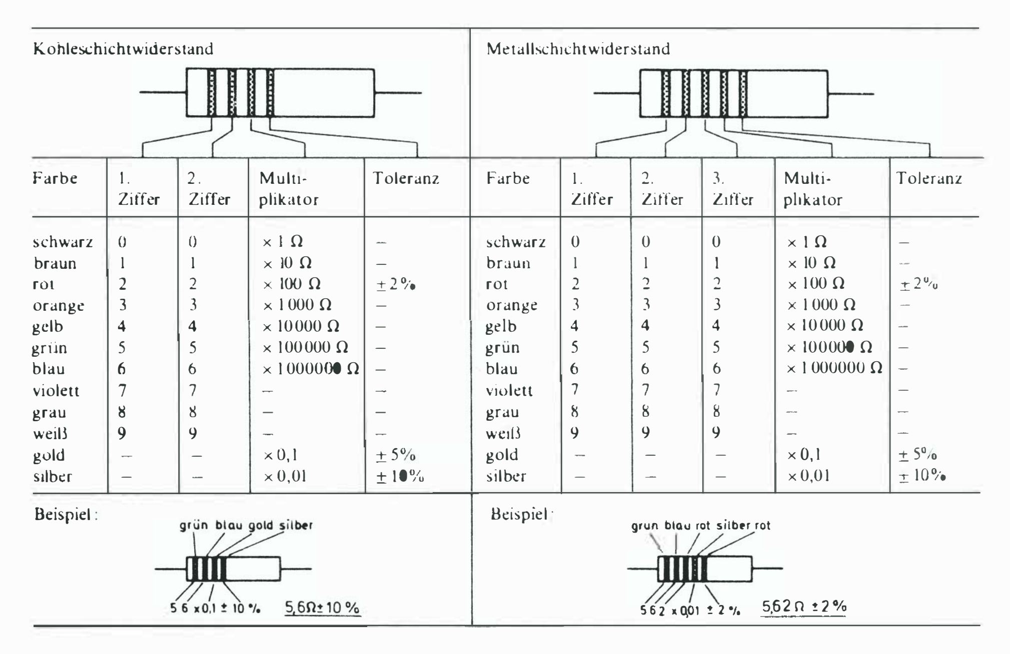
Dazu muß man wissen, daß elektronische Bauteile herstellungsbedingt immer eine gewisse Toleranz aufweisen. Ein Widerstand mit 330 Ohm und 10% Toleranz – dieser Wert ist üblich – kann damit Werte zwischen 297 und 363 Ohm haben. Es wäre

für die Händler nicht zumutbar, jedem Ohmwert eine eigene Schublade zuzuordnen. Außerdem kommt es in der Praxis nur bei sehr wenigen Anwendungen auf exakte Werte an. Gerade in der Digitaltechnik sind die Ansprüche meist äußerst gering, da normalerweise alle für die sichere Funktion verantwortlichen Teile bereits in den ICs untergebracht sind. Pull-Up-Widerstände (siehe Kapitel 2.2.4.2) sind beispielsweise völlig unkritisch. Ihr Wert kann durchaus zwischen etwa 400 Ohm bis 10 000 Ohm schwanken.
Die Bauteile werden in Normreihen eingeteilt, die alle möglichen Werte abdekken. Der Nennwert des nächsthöheren Widerstandswertes in der Normreihe E12 (12 Werte pro Dekade) mit ±10% Toleranz ist zum Beispiel 390 Ohm. Suchen Sie sich also den nächsten vorhandenen Wert aus.

Viele Baugruppen sind bereits mit aufeinander abgestimmten Einzelelementen in ein gemeinsames Gehäuse eingebaut. Musterbeispiel dafür ist natürlich das IC. Beinhaltet ein Gehäuse dagegen tatsächlich nur ein einziges Bauelement im elektrischen Sinn, dann spricht man von einem diskreten Bauelement. Im folgenden einige wichtige Angaben zu solchen universellen Elementen.

A.4.1 Widerstände

Wie Sie bereits wissen, werden Widerstände in Ohm (Ω) gemessen. Statt 1000 Ohm sagt man auch Kiloohm (kΩ) und statt 1 000 000 Ohm Megaohm (MΩ). Um verschiedene Werte unterscheiden zu können, kennzeichnet man sie mit vier oder fünf Farbringen. Bild A.1 zeigt deren Bedeu-



Kohleschichtwiderstand

Farbe	1. Ziffer	2. Ziffer	Multiplikator	Toleranz
schwarz	0	0	× 1 Ω	–
braun	1	1	× 10 Ω	–
rot	2	2	× 100 Ω	± 2%
orange	3	3	× 1 000 Ω	–
gelb	4	4	× 10 000 Ω	–
grün	5	5	× 100 000 Ω	–
blau	6	6	× 1 000000 Ω	–
violett	7	7	–	–
grau	8	8	–	–
weiß	9	9	–	–
gold	–	–	× 0,1	± 5%
silber	–	–	× 0,01	± 10%

Metallschichtwiderstand

Farbe	1. Ziffer	2. Ziffer	3. Ziffer	Multiplikator	Toleranz
schwarz	0	0	0	× 1 Ω	–
braun	1	1	1	× 10 Ω	–
rot	2	2	2	× 100 Ω	± 2%
orange	3	3	3	× 1 000 Ω	–
gelb	4	4	4	× 10 000 Ω	–
grün	5	5	5	× 100000 Ω	–
blau	6	6	6	× 1 000000 Ω	–
violett	7	7	7	–	–
grau	8	8	8	–	–
weiß	9	9	9	–	–
gold	–	–	–	× 0,1	± 5%
silber	–	–	–	× 0,01	± 10%

Bild A.1: Farbcode-Tabellen für Schichtwiderstände (IEC-Norm und DIN 41 429).

tung. Man liest die Ringe von demjenigen Anschluß her, dem sie am nächsten liegen. Metallfilmwiderstände, die in der Regel mit geringeren Toleranzen hergestellt werden als Kohleschichtwiderstände, tragen fünf Ringe. Damit lassen sich genauere Wertangaben machen.

Das Symbol für einen Widerstand ist ein Rechteck. In Schaltbildern, Bestückungsplänen und Bauteillisten werden Widerstandswerte oft abgekürzt. So hat es sich eingebürgert, das Zeichen für Ohm (Ω) einfach wegzulassen. Außerdem wird an die Stelle des Kommas häufig der Multiplikator k oder M gesetzt. Ein Widerstand von 5600 Ohm erscheint also zum Beispiel als 5,6k oder auch als 5k6.

1M5 wäre ein Widerstand mit 1,5 Megaohm – das sind 1 500 000 Ohm.

Widerstände gibt es nicht nur mit verschiedenen Ohmwerten, sondern auch für unterschiedliche Belastungen. In jedem Widerstand wird elektrische Leistung in Wärme umgewandelt. Je mehr Strom fließt, bzw. je höher die Spannung steigt, desto wärmer wird es dem Bauelement. Irgendwann aber haucht auch der eifrigste Widerstand sein Leben in Form von Rauchzeichen aus. Die dafür nötige Leistung nennt man die maximale Verlustleistung. Für unsere Schaltungen genügen Widerstände mit ¼ Watt Belastbarkeit, falls nicht anders gekennzeichnet. Natürlich können auch kräftigere Ausführungen gewählt werden. Diese sind dann allerdings teurer und brauchen mehr Platz.

In digitalen Schaltungen benötigt man oft mehrmals den gleichen Widerstandswert dicht nebeneinander, beispielsweise um das Potential eines offenen Eingangs mit einem Pull-up-Widerstand festzulegen. Dabei sind einseitig alle Einzelwiderstände auf eine gemeinsame Anschlußleitung geführt. Die benötigte Anordnung wird als SIL-Widerstandsnetzwerk gleich fertig verschaltet angeboten.

Außer festen Widerständen gibt es noch beliebig einstellbare, die sogenannten Potentiometer oder einfach Potis. Sie lassen sich zwischen 0 Ohm und dem aufgedruckten Maximalwert regeln. Trimmer sind ebenso aufgebaut, lassen sich aber nur mit einem Schraubenzieher verstellen.

A.4.2 Kondensatoren

Kondensatoren sind vom Prinzip her ganz einfache Bauelemente: zwei leitende Flächen, dazwischen eine Isolierschicht. Es gibt jedoch viele verschiedene Bauformen. In den meisten Anwendungen ist der Typ nicht maßgebend, so daß die Auswahl lediglich von den Kosten bestimmt wird.

Die Kapazität eines Kondensators wird in Farad (F) angegeben. Da 1F bereits ein ungewöhnlich hoher Wert ist, sind die Bezeichnungen μF (Mikrofarad = 0,000 F), nF (Nanofarad = 0,000 000 1 F) und pF (Pikofarad = 0,000 000 000 1 F) üblich.

Nach der Art ihres Aufbaus ist eine Einteilung in Folienkondensatoren, Keramikkondensatoren und Elektrolytkondensatoren möglich. Hier ein paar Hinweise auf besondere Merkmale und Unterschiede:

Folienkondensatoren werden aus mit Metallschichten bedampften Folien gewickelt. Es lassen sich sehr genaue Werte mit geringen Temperaturschwankungen erzielen (zum Beispiel Styroflex-Kondensatoren). Einige Ausführungen eignen sich jedoch nicht für Hochfrequenzanwendungen.

Keramikkondensatoren werden dagegen

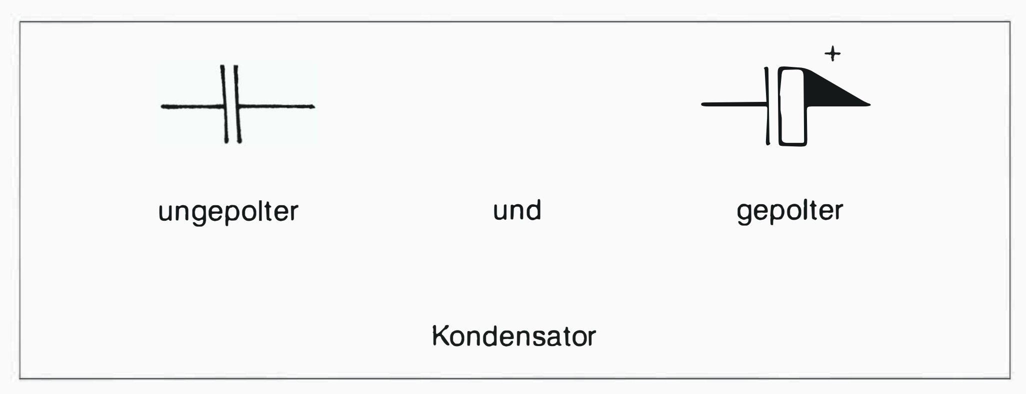


Bild A.2: Schaltsymbol eines Kondensators

hauptsächlich in der Hochfrequenztechnik eingesetzt. Auch für Digitalschaltungen sind sie zum Abblocken von sehr kurzen Spannungsspitzen besonders geeignet, um Störungen von der Betriebsspannung fernzuhalten. Kurzzeitige Spannungsschwankungen werden sozusagen durch die im Kondensator gespeicherte Ladung aufgefangen.

Solche Kondensatoren werden hauptsächlich im Piko- und Nanofaradbereich gebaut.

Elektrolytkondensatoren setzt man vorzugsweise bei sehr hohen Kapazitätswerten ein. Sie zeichnen sich durch eine relativ große Kapazität bei vergleichsweise geringer Baugröße und niedrigem Preis aus. Im Gegensatz zu den beiden anderen Bauformen sind »Elkos« jedoch gepolt und dürfen nicht an Wechselspannung betrieben werden. Bei falschem Einbau in die Platine wird ihre Isolierschicht zersetzt, was zu Kurzschluß und sogar zur Explosion führen kann! Die Polung des Bauteils ist daher in jedem Fall deutlich lesbar aufgedruckt. Bild A.2 zeigt die unterschiedliche Darstellungsweise eines gepolten und eines ungepolten Kondensators im Schaltplan. Der Pluspol wird durch ein Rechteck symbolisiert.

Bild A.3 zeigt die beiden wichtigsten Bauformen von Elkos. Unsere Platinen sind aus Platzgründen in der Regel für radiale Ausführungen vorgesehen. Bei axialen Typen ist der Minuspol zusätzlich mit einem umlaufenden Strich, der Pluspol dagegen mit einer ebensolchen Kerbe gekennzeichnet.

Elektrolytkondensatoren haben sehr große Toleranzen und verändern ihre Werte sogar mit der Zeit und mit den Betriebsbedingungen.

Im Vergleich zu den am häufigsten verwendeten Aluminium-Elektrolytkondensatoren besitzen die tropfenförmigen Tantal-Elektrolytkondensatoren bessere elektrische Eigenschaften und noch kleineren Raumbedarf. Sie sind jedoch ungleich teurer.

Die Kapazitätsangabe bei Kondensatoren ist mitunter etwas merkwürdig, und man muß schon ein wenig Phantasie besitzen und ungefähr den Wert abschätzen können, um sichere Aussagen machen zu kön-

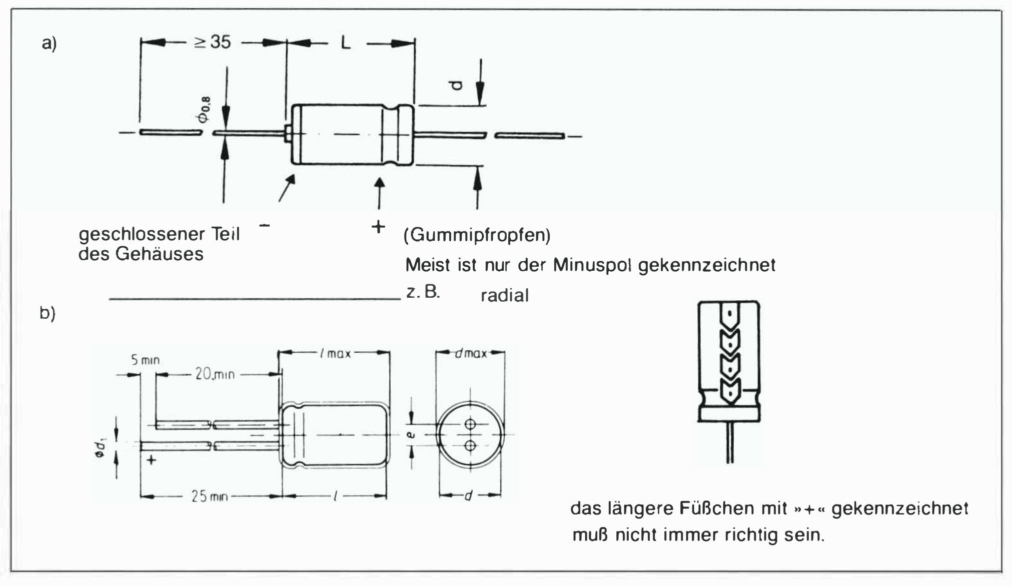


Bild A.3: Bauformen von axialen (a) und radialen (b) Elkos mit ihrer Polaritätskennzeichnung.

nen. Der Aufdruck .01 auf einem keramischen Scheibenkondensator bedeutet zum Beispiel 0,01 Mikrofarad, die Angabe 100 auf einem kleinen Keramik- oder Kunststofftyp dagegen 100 Pikofarad. 10/6 auf einem Elko heißt 10 Mikrofarad und maximal 6 Volt. Besonders auf japanischen Kondensatoren kann man alle möglichen Buchstaben sehen. Man sollte das nicht zu ernst nehmen. MF heißt nicht etwa Millifarad oder gar Megafarad (!), sondern es sind Mikrofarad gemeint. Oft sieht man dafür auch das Kürzel UF.

Jeder Kondensator ist für eine bestimmte maximale Spannung gebaut, die nicht überschritten werden darf. Andernfalls kann das Isoliermaterial zwischen den beiden leitenden Schichten zerstört werden. Diese maximale Spannung ist meist auf den Bauelementen angegeben. Sie liegt bei Folien- und Keramikkondensatoren so hoch, daß wir uns bei ihnen keine Gedanken machen müssen. Anders jedoch bei Elektrolytkondensatoren, die sich – je nach Spannungsfestigkeit – recht deutlich in Baugröße und Preis unterscheiden. Grundsätzlich kann man selbstverständlich Kondensatoren mit größeren Spannungsangaben als gefordert in die Schaltungen einbauen, man muß jedoch darauf achten, ob genügend Raum zum Einlöten auf der Platine vorhanden ist.

A.4.3 Halbleiter

Unter diesen Begriff fallen sehr viele unterschiedliche Bauelemente aus Halbleitermaterial (Germanium, Silizium . . .), zum Beispiel:

* Gleichrichterdioden
* Zenerdioden

* Fotodioden
* Leuchtdioden
* Kapazitätsdioden
* Transistoren
* Fototransistoren
* Operationsverstärker
* Brückengleichrichter
* Integrierte Schaltungen (ICs)
* Thyristoren
* Triacs
* Diacs
* Meßfühler usw.

Außer bei manchen Meßfühlern handelt es sich immer um gepolte Bauelemente, die bei falschem Einbau zerstört werden können. Achten Sie daher unbedingt auf die richtige Einbaulage nach Bestückungsplan. Die Bezeichnung ist bei jedem Bauteil aufgedruckt. Angaben über die jeweilige Anschlußbelegung entnehmen Sie bitte den zugehörigen Skizzen.
Für Einzelhalbleiter – auch diskrete Halbleiter genannt – gibt es umfangreiche Vergleichslisten. Bei den hier vorgestellten Schaltungen darf man ruhig darin aufgeführte entsprechende Vergleichstypen verwenden, falls nicht ausführlich davon abgeraten wird.

A.4.3.1 Transistoren

In der Digitaltechnik setzt man Transistoren vorrangig als Schalter ein, die durch eine elektrische Steuerspannung geöffnet und geschlossen werden. Der Schaltkontakt liegt zwischen Kollektor und Emitter, während die Steuerspannung an die Basis angelegt wird. Es genügt bereits ein kleiner Basisstrom, um den Schalter zu steuern. Dieser Sachverhalt wird in vielen Schaltungen, besonders in Kapitel 2, ausgenutzt.
Grundsätzlich gibt es zwei Arten von Transistoren, die nach ihrem inneren Aufbau in NPN- und PNP-Typen aufgeteilt werden. Ihre Unterschiede erläutert Kapitel 2.2.3.4.

A.4.3.2 Leuchtdioden

Leuchtdioden (LEDs = Light Emitting Diods) sind – wie normale Dioden (Bild A.4) auch – Halbleiterbauelemente. Das zur Herstellung verwendete Material bestimmt die Wellenlänge der ausgesandten elektromagnetischen Strahlung, was gleichbedeutend ist mit der Leuchtfarbe. Das Spektrum reicht dabei vom unsichtbaren infrarot über rot, gelb und grün bis blau. Dabei nimmt der Wirkungsgrad jedoch stetig ab und blauleuchtende LEDs sind zudem sehr teuer.
Die wichtigsten Vorteile von Leuchtdioden gegenüber Glühbirnchen sind der wesentlich geringere Leistungsbedarf, die ungleich längere Lebensdauer, die Unempfindlichkeit gegenüber mechanischen Stößen und vor allem die kurze Reaktionszeit beim Ein- und Ausschalten. Nach dem Einschalten der Spannung leuchtet eine LED

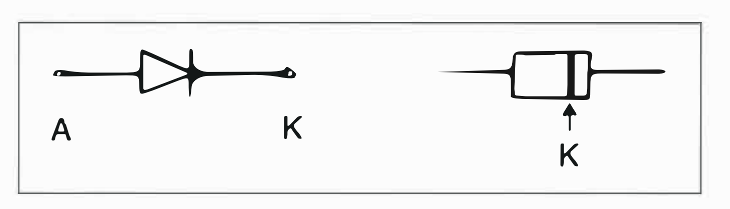


Bild A.4: Anschlußbelegung von Dioden (z.B. 1N4148, 1N4004...)

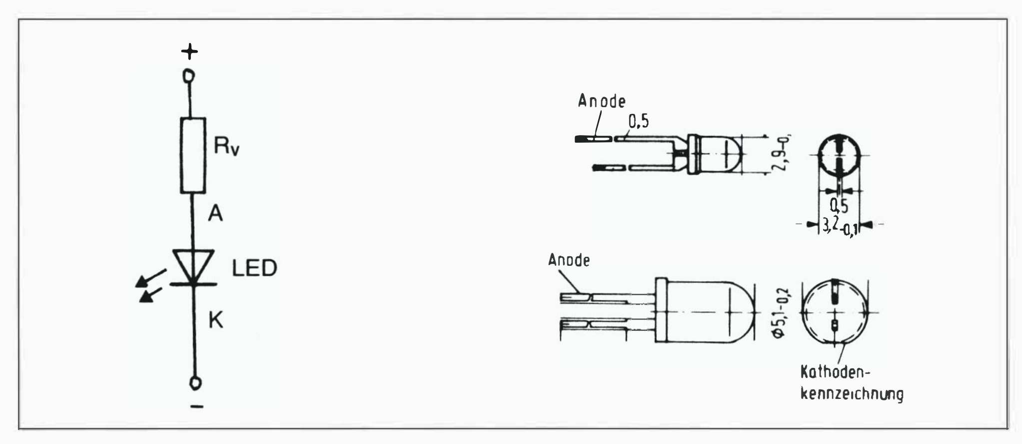


Bild A.5: Beschaltung und Anschlußbelegung von Leuchtdioden

bereits nach wenigen Tausendstelteilen einer Millionstelsekunde mit ihrer vollen Lichtstärke.

Leuchtdioden haben normalerweise zylindrische Plastikgehäuse nach Bild A.5 mit 3 mm oder 5 mm Durchmesser. Zur Kennzeichnung der Polung ist das Gehäuse an der Katodenseite meist abgeflacht. Am sichersten ist es, das Bauteil gegen das Licht zu halten. Die Silhouette mit der Anschlußzuordnung zeigt Bild A.5.

Eine Leuchtdiode darf nie ohne Vorwiderstand an die Spannung geschaltet werden, der den Strom begrenzt und damit die Helligkeit der LED einstellt. Bei 5 Volt sind Werte zwischen 330 Ohm und 1 Kiloohm üblich.

A.4.3.3 Integrierte Schaltungen

Natürlich kann hier bei der Vielzahl der erhältlichen ICs nicht auf jeden einzelnen Typ eingegangen werden. Es sei an dieser Stelle nur auf die Beschreibungen im Text hingewiesen.

Viele ICs werden von diversen Herstellern angeboten. Zur Kennzeichnung der Herkunft und spezieller Eigenschaften setzen manche Firmen bestimmte Buchstaben- oder Zahlenkombinationen vor und hinter die eigentliche Bezeichnung (siehe Anhang D). Man sollte sich von diesen Angaben nicht verwirren lassen.

Bei allen in diesem Buch vorgestellten Geräten können TTL-ICs der 74XX-Serie aus verschiedenen Familien genommen werden, falls nicht ausdrücklich davon abgeraten wird. Statt einem 74LS00 kann also durchaus auch ein 7400 oder ein 74HC00 Verwendung finden. Die Unterschiede werden in Kapitel 1.3.3 erläutert.

A.4.3.4 Die Behandlung von MOS-Bauteilen

MOS steht für den englischen Begriff Metall Oxide Semiconductor – zu deutsch: Metalloxid-Halbleiter. Gemeint ist eine bestimmte Fertigungsweise von elektronischen Halbleiterbauelementen. Diese Bauteile sind sehr empfindlich gegen statische Aufladung und falsches Einsetzen in die

Schaltung. Statische Ladung kann zum Beispiel entstehen, wenn Sie über einen Teppichboden gehen oder Kleidung mit Kunstfasern tragen. Durch Reibung laden Sie sich gegenüber Ihrer Umwelt elektrisch auf. Die so entstehende elektrische Spannung kann sehr hohe Werte annehmen. Berühren Sie nun ein MOS-IC, so wird dies mit hoher Wahrscheinlichkeit zerstört.

Deshalb sind beim Umgang mit MOS-Bauteilen folgende Vorsichtsmaßnahmen notwendig:

1. Sorgen Sie dafür, daß Ihr Körper nicht statisch geladen ist. Die statische Körperladung kann man durch Abreiben der Hände mit einem leitenden Material reduzieren.
2. Vermeiden Sie jeden direkten Kontakt mit den Anschlußpins, auch mit statisch aufladbarem Material, zum Beispiel Nylon. Dazu gehört auch die Lagerung der ICs in Kästchen aus Kunststoffmaterial.
3. Für den Transport sollten die Anschlußpins in einem leitenden Schaumstoff stecken, es sei denn, sie befinden sich bereits auf der Platine. Nur diese beiden Transportbedingungen halten die statischen Ladungen von den Anschlußpins fern.
4. Beim Einstecken der ICs in die Fassungen muß man auf ihre richtige Lage achten. Kennzeichen dafür ist die IC-Kerbe, wie bereits in Kapitel 1 erläutert.

A.4.4 Andere Bauelemente

Es gibt in der Elektronik natürlich noch eine Vielzahl anderer Elemente. Alle aufzuzählen, würde bei weitem den Rahmen dieses Buches sprengen. Spezielle Teile werden jeweils im Text der zugehörigen Bauanleitung genauer erklärt. Auf den folgenden Seiten finden Sie eine kurze Zusammenfassung der Pinbelegung von diversen Halbleitern.

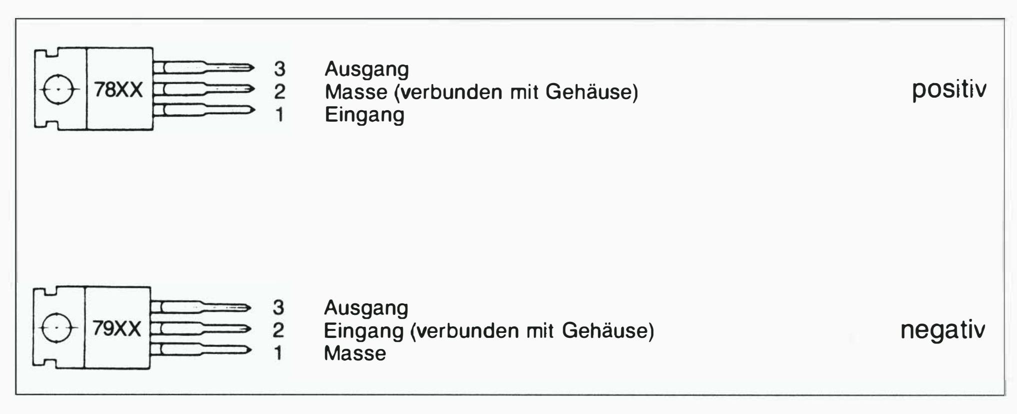


Bild A.6: Positiv- und Negativspannungsregler 78XX und 79XX

K A G

A_1 A_2 G

A

G

K

Thyristor

A_2

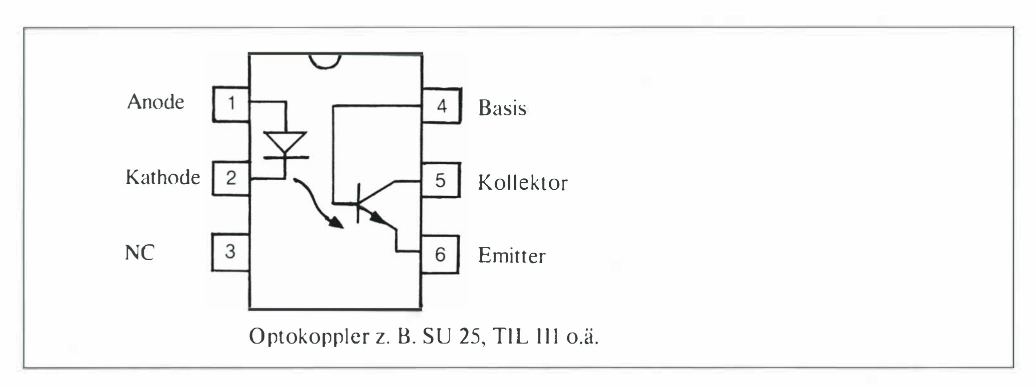
G

A_1

Triac

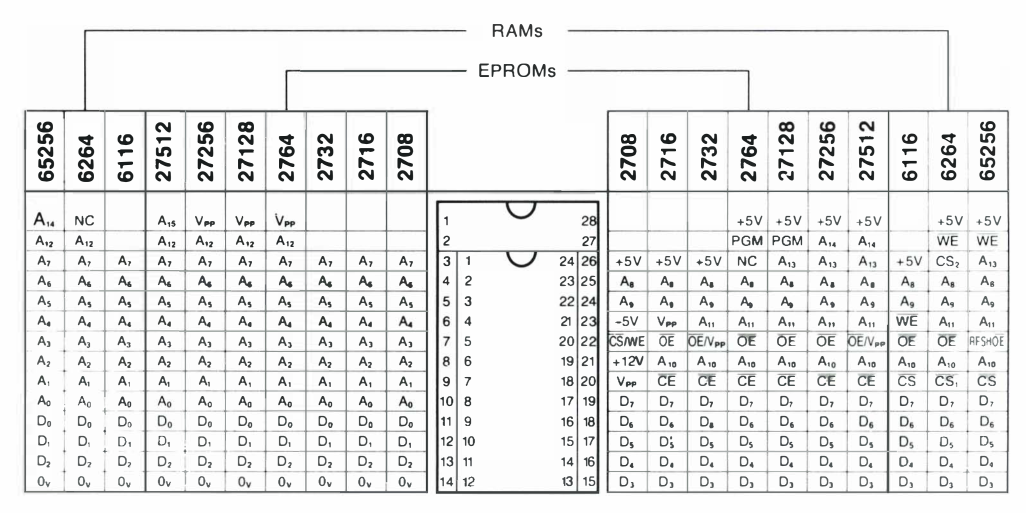
bei Thyristoren z. B. TYN 058... 1008 (8 A)
bei Triacs z. B. TIC 226 D (8 A)

Bild A.7: Schaltsymbole von Thyristor und Triac mit Pinbelegung im Gehäuse TO 220



Optokoppler z. B. SU 25, TIL 111 o.ä.

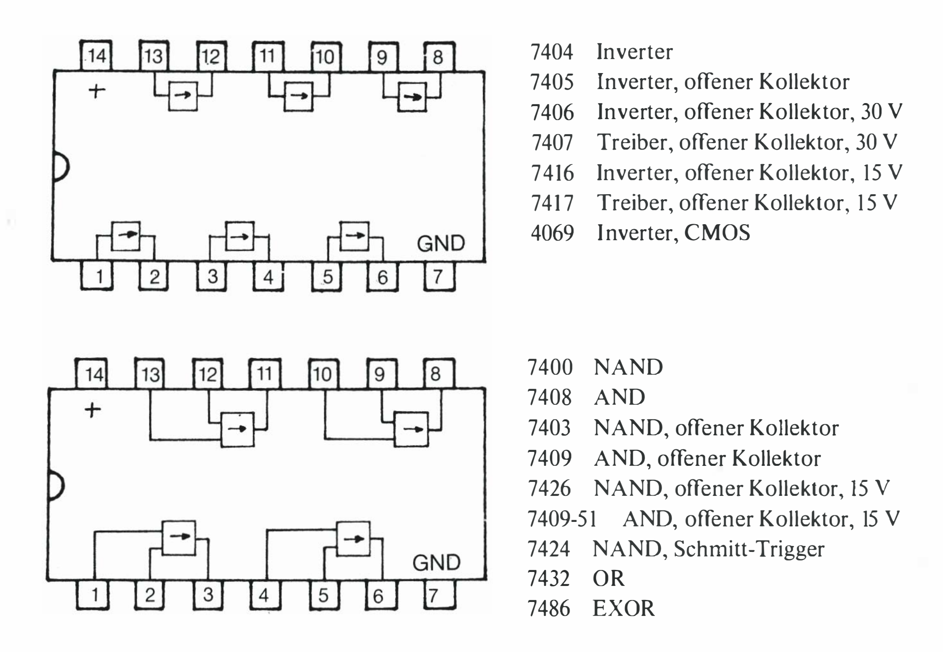
Bild A.8: Pinbelegung gängiger Optokoppler



65256	6264	6116	27512	27256	27128	2764	2732	2716	2708					2708	2716	2732	2764	27128	27256	27512	6116	6264	65256
A_{14}	NC		A_{15}	V_{PP}	V_{PP}	V_{PP}				1			28				+5V	+5V	+5V	+5V		+5V	+5V
A_{12}	A_{12}		A_{12}	A_{12}	A_{12}	A_{12}				2			27				PGM	PGM	A_{14}	A_{14}		$\overline{WE}$	$\overline{WE}$
A_7	A_7	A_7	A_7	A_7	A_7	A_7	A_7	A_7	A_7	3	1	24	26	+5V	+5V	+5V	NC	A_{13}	A_{13}	A_{13}	+5V	CS_2	A_{13}
A_6	A_6	A_6	A_6	A_6	A_6	A_6	A_6	A_6	A_6	4	2	23	25	A_8	A_8	A_8	A_8	A_8	A_8	A_8	A_8	A_8	A_8
A_5	A_5	A_5	A_5	A_5	A_5	A_5	A_5	A_5	A_5	5	3	22	24	A_9	A_9	A_9	A_9	A_9	A_9	A_9	A_9	A_9	A_9
A_4	A_4	A_4	A_4	A_4	A_4	A_4	A_4	A_4	A_4	6	4	21	23	-5V	V_{PP}	A_{11}	A_{11}	A_{11}	A_{11}	A_{11}	$\overline{WE}$	A_{11}	A_{11}
A_3	A_3	A_3	A_3	A_3	A_3	A_3	A_3	A_3	A_3	7	5	20	22	$\overline{CS}$/WE	$\overline{OE}$	$\overline{OE}/V_{PP}$	$\overline{OE}$	$\overline{OE}$	$\overline{OE}$	$\overline{OE}/V_{PP}$	$\overline{OE}$	$\overline{OE}$	RFSH $\overline{OE}$
A_2	A_2	A_2	A_2	A_2	A_2	A_2	A_2	A_2	A_2	8	6	19	21	+12V	A_{10}	A_{10}	A_{10}	A_{10}	A_{10}	A_{10}	A_{10}	A_{10}	A_{10}
A_1	A_1	A_1	A_1	A_1	A_1	A_1	A_1	A_1	A_1	9	7	18	20	V_{PP}	$\overline{CE}$	$\overline{CE}$	$\overline{CE}$	$\overline{CE}$	$\overline{CE}$	$\overline{CE}$	$\overline{CS}$	$\overline{CS_1}$	$\overline{CS}$
A_0	A_0	A_0	A_0	A_0	A_0	A_0	A_0	A_0	A_0	10	8	17	19	D_7	D_7	D_7	D_7	D_7	D_7	D_7	D_7	D_7	D_7
D_0	D_0	D_0	D_0	D_0	D_0	D_0	D_0	D_0	D_0	11	9	16	18	D_6	D_6	D_6	D_6	D_6	D_6	D_6	D_6	D_6	D_6
D_1	D_1	D_1	D_1	D_1	D_1	D_1	D_1	D_1	D_1	12	10	15	17	D_5	D_5	D_5	D_5	D_5	D_5	D_5	D_5	D_5	D_5
D_2	D_2	D_2	D_2	D_2	D_2	D_2	D_2	D_2	D_2	13	11	14	16	D_4	D_4	D_4	D_4	D_4	D_4	D_4	D_4	D_4	D_4
0_V	0_V	0_V	0_V	0_V	0_V	0_V	0_V	0_V	0_V	14	12	13	15	D_3	D_3	D_3	D_3	D_3	D_3	D_3	D_3	D_3	D_3

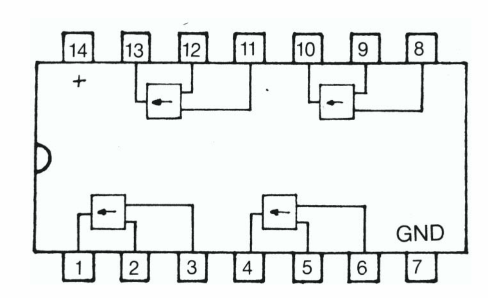
Bild A.9: Die Pinbelegung der wichtigsten Speicherbausteine

Die nachfolgenden Bilder geben Auskunft über die Pinbelegung von Logik-Bausteinen:

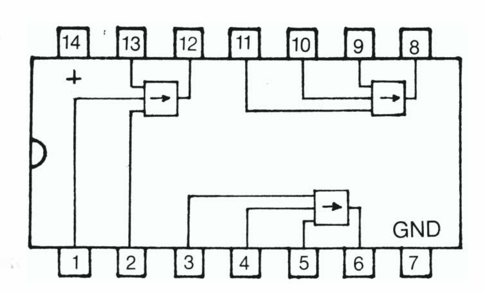


7404 Inverter
7405 Inverter, offener Kollektor
7406 Inverter, offener Kollektor, 30 V
7407 Treiber, offener Kollektor, 30 V
7416 Inverter, offener Kollektor, 15 V
7417 Treiber, offener Kollektor, 15 V
4069 Inverter, CMOS

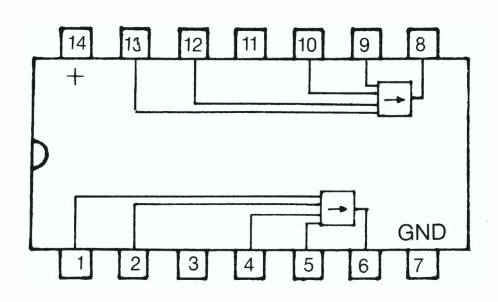
7400 NAND
7408 AND
7403 NAND, offener Kollektor
7409 AND, offener Kollektor
7426 NAND, offener Kollektor, 15 V
7409-51 AND, offener Kollektor, 15 V
7424 NAND, Schmitt-Trigger
7432 OR
7486 EXOR



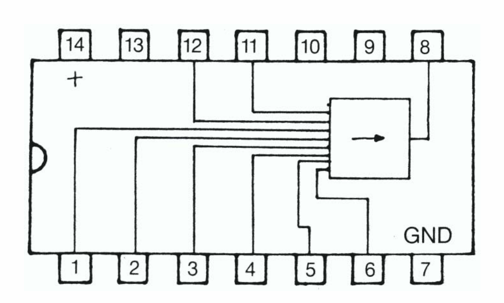
7402 NOR
7428 NOR, für hohe Ausgangslast
7433 NOR, offener Kollektor



7410 NAND
7411 AND
7412 NAND, offener Kollektor
7415 AND, offener Kollektor
7427 NOR



7420 NAND
7421 AND
7422 NAND, offener Kollektor, 5,5 V
7440 NAND, für hohe Ausgangslast
7413 NAND, Schmitt-Trigger
7418 NAND, Schmitt-Trigger



7430 NAND

Anhang B
Register des CIA 6526

Register	Adresse	Verwendung im C 64	Funktion bei Schreiben	Funktion bei Lesen	Beschreibung in
0	$DC00 = 56320 $DD00 = 56576	Tastaturabfrage ser. Bus, RS232	PRA (Port Register A) Programmieren der Ausgangspegel LOW bei jew. Bit=0, HIGH bei Bit=1	Eingänge: Pegel der Anschlüsse	2.2.1.2 2.2 2.4.1.3
1	$DC01 = 56321 $DD01 = 56577	Tastaturabfrage User-Port/RS232	PRB (Port Register B) Programmieren der Ausgangspegel LOW bei jew. Bit=0, HIGH bei Bit=1	Eingänge: Pegel der Anschlüsse	2.2.1.2 2.2 2.4.1.3
2	$DC02 = 56322 $DD02 = 56578	8 Ausgänge (Spalten) normal 63 = 00111111	DDRA (Data Direction Register A = Datenrichtungsregister A) Programmieren als Ein-/Ausgänge Eingang bei jew. Bit = 0, Ausgang bei Bit=1	Lesen derE/A-Programmierung	2.2.1.1 2.2 2.4.1.3
3	$DC03 = 56323 $DD03 = 56579	8 Eingänge (Zeilen) normal Eingänge	DDRB (Data Direction Register A = Datenrichtungsregister B) Programmieren als Ein-/Ausgänge Eingang bei jew. Bit=0, Ausgang bei Bit=1	Lesen der E/A-Programmierung	2.2.1.1 2.2 2.4.1.3
4	$DC04 = 56324 $DD04 = 56580	IRQ (alle 1/60 s) +RS232	TA LO (Timer A LOW-Byte) Startwert für Abwärtszählen	Augenblicklicher Zählerstand	2.4.1
5	$DC05 = 56325 $DD05 = 56581	IRQ (alle 1/60 s) RS232	TA HI (Timer A HIGH-Byte) Startwert für Abwärtszählen	Augenblicklicher Zählerstand	2.4.1
6	$DC06 = 56326 $DD06 = 56582	Kassettenbetrieb RS232	TB LO (Timer B LOW-Byte) Startwert für Abwärtszählen	Augenblicklicher Zählerstand	2.4.1
7	$DC07 = 56327 $DD07 = 56583	Kassettenbetrieb RS232	TB HI (Timer B HIGH-Byte) Startwert für Abwärtszählen	Augenblicklicher Zählerstand	2.4.1
8	$DC08 = 56328 $DD08 = 56584	für RND unbenutzt	TOD 10THS (Time Of Day = Echtzeituhr 1/10 Sekunden) Bit 0-3: 1/10 Sekunden BCD-Format Bit 4-7 immer = 0 Reg. 15, Bit 7: Uhr-/Alarmzeit	Lesen des 1/10 s-Registers immer aktuelle Uhrzeit	3.4.7 3.4.10

Register	Adresse	Verwendung im C64	Funktion bei Schreiben	Funktion bei Lesen	Beschreibung in
9	$DC09 = 56329 $DD09 = 56585	für RND unbenutzt	TOD SEC (Time Of Day = Echtzeituhr Sekunden) Stellen Bit 0-3: Sekunden Einer Bit 4-7: Sekunden Zehner Reg. 15, Bit 7: Uhr-/Alarmzeit	Lesen Bit 0-3: Sekunden Einer Bit 4-7: Sekunden Zehner immer aktuelle Uhrzeit	3.4.7 3.4.10
10	$DC0A = 56330 $DD0A = 56586	für RND unbenutzt	TOD MIN (Time Of Day = Echtzeituhr Minuten) Stellen Bit 0-3: Minuten Einer Bit 4-7: Minuten Zehner Reg. 15, Bit 7: Uhr-/Alarmzeit	Lesen Bit 0-3: Minuten Einer Bit 4-7: Minuten Zehner immer aktuelle Uhrzeit	3.4.7 3.4.10
11	$DC0B = 56331 $DD0B = 56587	für RND unbenutzt	TOD HR (Time Of Day = Echtzeituhr Stunden) Stellen Bit 0-3: Stunden Einer Bit 4: Stunden Zehner Bit 7: AM=0, PM=1 Reg. 15, Bit 7: Uhr-/Alarmzeit	Lesen Bit 0-3: Stunden Einer Bit 4: Stunden Zehner Bit 7: 0=AM, 1=PM immer aktuelle Uhrzeit	3.4.7 3.4.10
12	$DC0C = 56332 $DD0C = 56588	unbenutzt unbenutzt	SDR (Serial Data Register = Serielles Datenregister) Schieberegister für Pin SP Ausg. Höchstwertiges Bit erscheint zuerst	Schieberegister für Pin SP Eing.	2.5.1
13	$DC0D = 56333 $DD0D = 56589		ICR (Interrupt Control Register = Interrupt-Kontrollregister) Interruptmaskierung für Funktion: Bit 0: Unterlauf Timer A aufgetreten Bit 1: Unterlauf Timer B aufgetreten Bit 2: Echtzeituhr hat Alarmzeit erreicht Bit 3: Shiftregister voll (Eingang) bzw. leer (Ausgang) Bit 4: negative Flanke am Pin $\overline{FLAG}$ aufgetreten Bit 5 – 6: immer 0 Bit 7 = 1: 1-Bits machen scharf = 0: 1-Bits entschärfen Nur scharfe Bits erzeugen Interrupt	jew. Bit=1 falls Bedingung wahr 1 = mindestens eine Bedingung bei scharfem Maskenbit aufgetreten Bits werden bei Lesen gelöscht!	2.4.1.4 2.4.1.3 2.4.1.3 2.5.1 2.3.6.8 2.3.1 2.4.1.4
14	$DC0E = 56334 $DD0E = 56590		CRA (Control Register A = Kontrollregister A) Bit 0: 0: Timer A Stop, 1: Timer A Start Bit1: 1: Signalisierung von Unterlauf Timer A an PB6 wie folgt: Bit 2: 0: jeder Unterlauf von A kippt PB6 in die andere Lage 1: jeder Unterlauf erzeugt an PB6 einen HIGH-Impuls von 1μs Bit 3: 0: Timer A zählt fortlaufend vom Ausgangswert (Reg.4/5) abw. 1: Timer A zählt nur einmal vom Ausgangswert (Reg.4/5) auf 0 Bit 4: 0: keine Funktion 1: Register 4/5 wird als Startwert für Timer A übernommen Bit 5: Timertrigger: 0: Timer A zählt Systemtaktpulse 1: Timer A zählt steigende Flanken an Pin CNT Bit 6: 0: SP ist Schieberegistereingang, 1: SP ist Ausgang Bit 7: Echtzeituhrtrigger: 0: 60 Hz, 1: 50 Hz		2.4.1.2 2.4.1.3 2.4.4.5.1 2.4.2.1 2.5.1 3.4.10

Register	Adresse	Verwendung im C64	Funktion bei Schreiben	Funktion bei Lesen	Beschreibung in
15	$DC0F = 56335 $DD0F = 56591		CRB (Control Register B = Kontrollregister B) Bit 0: 0: Timer B Stop, 1: Timer B Start Bit 1: 1: Signalisierung von Unterlauf Timer B an PB7 wie folgt: Bit 2: 0: jeder Unterlauf von B kippt PB7 in die andere Lage 1: jeder Unterlauf erzeugt an PB7 einen HIGH-Impuls von 1μs Bit 3: 0: Timer B zählt fortlaufend vom Ausgangswert (Reg.6/7) abw. 1: Timer B zählt nur einmal vom Ausgangswert (Reg.6/7) auf 0 Bit 4: 0: keine Funktion 1: Register 6/7 wird als Startwert für Timer B übernommen Bit 5, 6: Timertrigger: 00: Timer B zählt Systemtaktpulse 01: Timer B zählt steigende CNT-Flanken 10: Timer B zählt Unterläufe von Timer A 11: Timer B zählt nur dann Unterläufe von A, wenn Pin CNT HIGH ist Bit 7: Schreiben in Reg. 9-11 stellt bei 0: Uhrzeit, 1: Alarmzeit		2.4.1.2 2.4.1.3 2.4.4.5.1 2.4.2.1

Anhang B: Register des CIA 6526

Anhang C
Pinbelegung der Steckverbindungen am C64

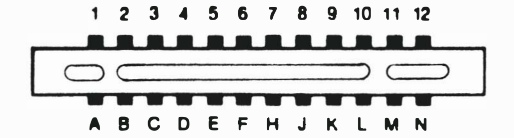
Dieser Anhang enthält die Pinbelegung aller Stecker am C64, angefangen mit dem User-Port ganz links hinten bis zum Control-Port 1 ganz rechts außen. Dabei wurde die Belegung immer so dargestellt, wie man sie sieht, wenn man von außen her auf die Kontakte schaut. Besonders beim Expansion-Port ist diese Angabe wichtig, da er in alten Handbüchern falsch dargestellt wurde.

In der Regel sind die Stecker verpolungssicher, doch Vorsicht mit Platinendirektsteckern! Bei unmittelbar an den Stecker gelöteten Platinen dürfte es schon auffallen, wenn sich die Bauteile auf der Unterseite befinden, doch besonders beim Einsatz von Kabeln ist es leicht möglich, den Stecker mit der Unterseite nach oben aufzustecken. Solche Irrtümer können schwerwiegende Folgen haben, die sich aber leicht vermeiden lassen.

Sowohl der User-Port- wie auch der Kassettenanschluß haben Einfräsungen zwischen bestimmten Kontaktpins der Platine. Setzt man in den Stecker an diesen Stellen Kodierstege ein, lassen sich die Stecker nicht mehr verkehrt herum aufstecken. Als Kodierstege können sehr gut breite Lötstützpunkte dienen, die in jedem Elektronikgeschäft zu haben sind. Sie werden mit einer Flachzange an der vorgesehenen Stelle im Stecker bis zum Anschlag eingepreßt und ihr herausragendes Ende anschließend mit einem Saitenschneider plan abgeschnitten. Verpolen ist nun unmöglich.

Denken Sie bitte daran, die Stecker nur bei ausgeschaltetem Computer aufzustecken bzw. abzuziehen.

Der User-Port

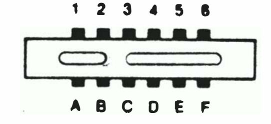


Pin	Bezeichnung	Bedeutung	Richtung	Erläuterung	Beschreibung in
1	GND	Ground (= Masse)	–	Systemmasse (0 Volt)	1.3.2
2	+5V DC	Betriebsspannung	–	Versorgungsspannung (+5V) für Erweiterungen	7.5
3	$\overline{\text{RESET}}$	Reset (= Rückstellen)	Ein/Aus	Zieht man diese Leitung auf LOW, dann werden alle Bausteine neu initialisiert. Der Programmzähler der CPU wird mit dem Inhalt der Zellen $FFFC und $FFFD geladen, der normalerweise auf $FCE2 zeigt: die Reset-Routine.	6.4.4 2.2.2
4	CTN1	Count (= Zählen)	Progrm.	Anschluß von CIA # 1. Kann vielseitig benutzt werden für den Zählerbetrieb und als Takt-Ein-/Ausgang beim seriellen Port.	2.5.1
5	SP1	Serieller Port	Progrm.	Serieller Ein-/Ausgang des Schieberegisters von CIA # 1.	2.5.1
6	CNT2	Count (= Zählen)	Progrm.	Anschluß von CIA # 2. Kann vielseitig benutzt werden für den Zählerbetrieb und als Takt-Ein-/Ausgang beim seriellen Port.	2.5.1
7	SP2	Serieller Port	Progrm.	Serieller Ein-/Ausgang des Schieberegisters von CIA # 2.	2.5.1
8	$\overline{\text{PC2}}$	Port Control	Ausgang	Anschluß von CIA # 2. Wird nach einem Schreib- oder Lesezugriff auf das Portregister B für eine Taktperiode LOW. Ausgang mit offenem Kollektor, also Pull-Up-Widerstand nötig!	2.3.1
9	SER. ATN	Attention vom seriellen Bus	Ein/Aus	Attention-Leitung des seriellen Bus. Kann als Ausgang vom Computer benutzt werden (invertiertes PA3 von CIA # 2). Über Open-Collector-Ausgang ist aber auch Beeinflussung dieser Busleitung möglich.	4.2 4.3
10 11	9 VAC 9 VAC	9-Volt-Wechselspannung	–	Direkt mit dem Transformator verbunden. Belastbarkeit ca. 100 mA. Dient zur vielseitigen Spannungserzeugung.	7.1 7.2
12	GND	Ground (= Masse)	–	Systemmasse (0 Volt)	1.3.2

Pin	Bezeichnung	Bedeutung	Richtung	Erläuterung	Beschreibung in
A	GND	Ground (= Masse)	–	Systemmasse (0 Volt)	1.3.2
B	FLAG2	Flag-Flipflop CIA # 2	Eingang	Eine negative Flanke an diesem Anschluß setzt das FLAG-Bit im Interrupt-Kontrollregister von CIA # 2.	2.3.1
C D E F H J K L	PBO PBl PB2 PB3 PB4 PB5 PB6 PB7	Portbit B Ø Portbit B 1 Portbit B 2 Portbit B 3 Portbit B 4 Portbit B 5 Portbit B 6 Portbit B 7	Progrm. Progrm. Progrm. Progrm. Progrm. Progrm. Progrm. Progrm.	Portregister B von CIA # 2. Jede Leitung ist einzeln auf Ein- oder Ausgabe programmierbar.	2.2
M	PA2	Portbit A 2	Progrm.	Portregister A, Bit 2 von CIA # 2.	2.2
N	GND	Ground (= Masse)	–	Systemmasse (0 Volt)	1.3.2

Pinbelegung des User-Ports

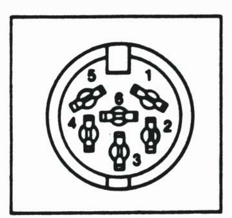
Der Kassetten-Port



Pin	Bezeichnung	Bedeutung	Richtung	Erläuterung	Beschreibung in
A-1	GND	Ground (= Masse)	–	Systemmasse (0 Volt)	1.3.2
B-2	+5V DC	Betriebsspannung	–	Versorgungsspannung (+5 Volt) für Erweiterungen	7.5
C-3	CASS. MOTOR	Motorspannung für die Datasette	Ausgang	Gesteuert von P5 der CPU (Adresse 1, Bit 5). Ausgangsspannung etwa 6V, stabilisiert bei P5=LOW; 0V bei HIGH.	3.1 3.4.3
D-4	CASS. READ	Lesesignale vom Band	Ein/Aus	Liegt mit SRQ des seriellen Bus zusammen über einen Pull-Up-Widerstand (3K3) an FLAG von CIA # 1.	3.1 2.3.1
E-5	CASS. WRITE	Schreibsignale vom Band	Progrm.	Liegt an P3 des Prozessor-Port (Adresse 1, Bit 3). Normalerweise ist diese Leitung ein Ausgang.	3.1
F-6	CASS. SENSE	Signal von der Datasette	Progrm.	Liegt über Pull-Up-Widerstand (3K3) an P4 der CPU. Ist normalerweise Eingang und meldet bei angeschlossener Datasette mit LOW, daß eine Taste gedrückt ist.	3.1

Pinbelegung des Kassetten-Port

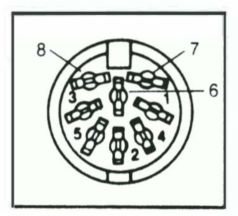
Der serielle Bus



Pin	Bezeichnung	Bedeutung	Richtung	Erläuterung	Beschreibung in
1	SER $\overline{\text{SRQ}}$	Service Request (Bedienungs-anforderung)	Ein/Aus	Liegt mit CASS. READ vom Kasset-ten-Port zusammen über einen Pull-Up-Widerstand (3K3) an $\overline{\text{FLAG}}$ von CIA # 1.	4.2 4.5
2	GND	Ground (= Masse)	–	Systemmasse (0 Volt)	1.3.2
3	SER. $\overline{\text{ATN}}$	Attention (= Achtung)	Ein/Aus	Wird diese Leitung LOW, unter-brechen alle angeschlossenen Geräte ihre Arbeit und widmen sich dem Busbetrieb. Beeinflußbar durch PA3 von CIA # 2 (negiert).	4.2 4.3
4	SER. CLK	Clock (= Übertragungs-(takt)	Ein/Aus	Geht bei Gültigkeit des Datenbit von LOW nach HIGH. Beeinflußbar durch PA4 von CIA # 2 (negiert) und lesbar über PA6 von CIA # 2 (nicht negiert).	4.2 4.3
5	SER. DATA	Serielle Daten	Ein/Aus	Beeinflußbar durch PA5 von CIA # 2 (negiert) und lesbar über PA7 von CIA # 2 (nicht negiert).	4.2 4.3
6	$\overline{\text{RESET}}$	Reset (= Rück-stellen)	Ein/Aus	Zieht man diese Leitung auf LOW, dann werden alle Bausteine neu initialisiert. Der Programmzähler der CPU wird mit dem Inhalt der Zellen \$FFFC und \$FFFD geladen, der normalerweise auf \$FCE2 zeigt: die Reset-Routine.	6.4.4 4.2 4.3 4.4

Pinbelegung des seriellen Bus

Die Audio-/Video-Buchse



Pin	Bezeichnung	Bedeutung	Richtung	Erläuterung	Beschreibung in
1	LUMINANCE	Video-Y-Signal	Ausgang	Bildhelligkeitssignal für Monitore vom Videocontroller (BAS-Signal)	5.1 5.2
2	GND	Ground (= Masse)	–	Systemmasse (0 Volt)	1.3.2
3	AUDIO OUT	Tonsignal	Ausgang	Tonsignal von Sound-Chip über einen Transistor ausgekoppelt. Spannung etwa 2V-Spitze-Spitze.	5.3
4	VIDEO OUT	Video-Mischsignal	Ausgang	Gemischtes Farb- und Helligkeitssignal vom Video-Chip (FBAS-Signal).	5.2
5	AUDIO IN	Tonsignal	Eingang	Ist über einen Kondensator an den Erweiterungseingang des Sound-Chip gelegt. Die Signalamplitude darf 3V-Spitze-Spitze nicht überschreiten!	
6	CHROMINANCE	Video-Farbsignal	Ausgang	An diesem Pin steht das Farbsignal separat zur Verfügung. Es wird vor allem beim Commodore-Farbmonitor benötigt und ermöglicht eine bessere Bildqualität.	5.2

Pinbelegung der Audio-/Video-Buchse

Der Expansion-Port

22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1

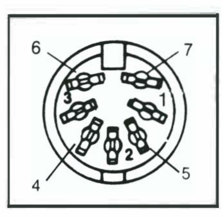
Z Y X W V U T S R P N M L K J H F E D C B A

Pin	Bezeichnung	Bedeutung	Richtung	Erläuterung	Beschreibung in
1	GND	Ground (= Masse)	–	Systemmasse (0 Volt)	1.3.2
2 3	+5V DC +5V DC	Betriebsspannung	–	Versorgungsspannung (+5 Volt) für Erweiterungen Maximale Belastung ca. 450mA	7.5
4	$\overline{IRQ}$	Interrupt Request (= Unter-brechungs-anforderung)	Eingang	Bei LOW-Pegel an diesem Anschluß bearbeitet die CPU zunächst den laufenden Befehl noch fertig, merkt sich dessen Position im Stack und lädt den Programmzähler mit dem Inhalt der Zellen $FFFE und $FFFF. Normalerweise zeigt dieser Vektor auf $FF48, die IRQ-Routine des Betriebs-systems.	2.3.6.8
5	R/$\overline{W}$	Read/$\overline{Write}$	Ausgang	Führt bei einem Lesezyklus HIGH, beim Schreiben LOW	6.1
6	DOTCLOCK	Bildpunktfrequenz	Ausgang	Taktfrequenz des Videoprozessors (bei PAL, 7,80 MHz)	6.5
7	$\overline{I/O1}$	Input/Output 1 (= Ein-/Ausgabe)	Ausgang	Nur LOW, falls eine Bitkombination zwischen $DE00 und $DEFF auf dem Adreßbus liegt	6.3.3
8	$\overline{GAME}$	Spielkonfiguration	Eingang	Dient zur Speicher-Umkonfigurierung (ROM-Einblendung)	6.4.2
9	$\overline{EXROM}$	Externes ROM	Eingang	Ähnlich $\overline{GAME}$. Bei LOW wird das interne RAM im Bereich $8000 – $9FFF ausgeschaltet und statt dessen an $\overline{ROML}$ LOW ausgegeben, falls eine dieser Adressen auftritt.	6.4.2
10	$\overline{I/O2}$	Input/Output 2 (= Ein-/Ausgabe)	Ausgang	Nur LOW, falls eine Bitkombination zwischen $DF00 und $DFFF auf dem Adreßbus liegt	6.3.3
11	$\overline{ROML}$	ROM low	Ausgang	Wird LOW, falls der Bereich $8000 bis $8FFF mit $\overline{EXROM}$ abgeschaltet ist und eine dieser Adressen auftritt.	6.4.2
12	BA	Bus Available (= Bus verfügbar)	Ausgang	Signal von Videocontroller, das die Gültigkeit der Busdaten anzeigt. Bei BA=1 belegt der VIC den Bus und keine andere Einheit darf ihn benutzen.	6.5
13	$\overline{DMA}$	Direkt Memory Access (= Direkter Speicherzugriff)	Eingang	Durch $\overline{DMA}$ = LOW kann der Prozessor aufgefordert werden, den Bus freizugeben. Er hält nach dem nächsten Lesezyklus an und alle Busleitungen werden hochohmig. So können andere Einheiten die Computerhardware benutzen. Bei $\overline{DMA}$ = HIGH arbeitet die CPU weiter.	

Pin	Bezeichnung	Bedeutung	Richtung	Erläuterung	Beschreibung in
14 15 16 17 18 19 20 21	CD7 CD6 CD5 CD4 CD3 CD2 CD1 CD0	CPU-Datenbit 7 CPU-Datenbit 6 CPU-Datenbit 5 CPU-Datenbit 4 CPU-Datenbit 3 CPU-Datenbit 2 CPU-Datenbit 1 CPU-Datenbit 0	Ein/Aus je nach Zyklus	Datenbus des Rechners. Das C steht für CPU. Da andere Datenleitungen sowieso nicht am Port zugänglich sind, wird das C meist weggelassen.	6.1
22	GND	Ground (= Masse)	—	Systemmasse (0 Volt)	1.3.2
A	GND	Ground (= Masse)	–	Systemmasse (0 Volt)	1.3.2
B	$\overline{ROMH}$	ROM high	Ausgang	Wird je nach Beschaltung von $\overline{GAME}$ und $\overline{EXROM}$ bei Zugriffen auf eine Adresse zwischen $A000 – $BFFF bzw. $E000 – $FFF LOW, falls der jeweilige interne Bereich ausgeblendet ist.	6.4.2
C	$\overline{RESET}$	Reset (= Rückstellen)	Ein/Aus	Zieht man diese Leitung auf LOW, dann werden alle Bausteine neu initialisiert. Der Programmzähler der CPU wird mit dem Inhalt der Zellen $FFFC und $FFFD geladen, der normalerweise auf $FCE2 zeigt: die Reset-Routine.	6.4.4 6.4.1 2.2.2
D	$\overline{NMI}$	Non Maskable Interrupt (= Nicht abschaltbare Unterbrechung)	Eingang	Bei LOW-Pegel an diesem Anschluß bearbeitet die CPU zunächst den laufenden Befehl noch fertig, merkt sich dessen Position im Stack und lädt den Programmzähler mit dem Inhalt der Zellen $FFFA und $FFFB. Normalerweise zeigt dieser Vektor auf $FE43, die NMI-Routine des Betriebssystems.	2.3.6.8
I	∅2	Phi 2 (Systemtakt)	Ausgang	Zeitraster für die Programmbearbeitung (ca. 0,98MHz)	6.3.2
F H J K L M N P R S T U V W X Y	CA15 CA14 CA13 CA12 CA11 CA10 CA9 CA8 CA7 CA6 CA5 CA4 CA3 CA2 CA1 CA0	CPU-Adreßbit 15 CPU-Adreßbit 14 CPU-Adreßbit 13 CPU-Adreßbit 12 CPU-Adreßbit 11 CPU-Adreßbit 10 CPU-Adreßbit 9 CPU-Adreßbit 8 CPU-Adreßbit 7 CPU-Adreßbit 6 CPU-Adreßbit 5 CPU-Adreßbit 4 CPU-Adreßbit 3 CPU-Adreßbit 2 CPU-Adreßbit 1 CPU-Adreßbit 0	Ausgang	Adreßbus des Computers. Das C steht für CPU. Da andere Adreßanschlüsse (zum Beispiel die des Videobereichs) sowieso nicht am Port zugänglich sind, wird das C meist weggelassen.	6.1
Z	GND	Ground (= Masse)	–	Systemmasse (0 Volt)	1.3.2

Pinbelegung des Expansion-Ports

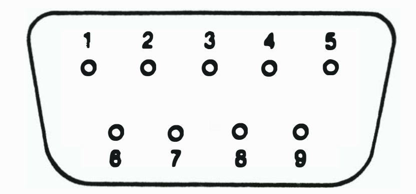
Der Netzstecker



Pin	Bezeichnung	Bedeutung	Richtung	Erläuterung	Beschreibung in
1 2 3	GND	Ground (= Masse)	–	Systemmasse (0 Volt) und Zuleitungs-abschirmung	1.3.2
4	N.C.	Not Connected	–	(nicht belegt)	1.3.4
5	+5V DC	Betriebsspannung	–	Versorgungsspannung (+ 5 Volt) für den C 64 und alle Erweiterungen. Belastbarkeit beim Original-Netzteil insgesamt 1,5A.	7.5
6 7	9 VAC 9 VAC	9-Volt-Wechsel-spannung	–	Versorgungsspannung für internes Netzteil und Erweiterungen am User-Port. Belastbarkeit insgesamt beim Original-Netzteil etwa 1A.	7.1 7.2

Pinbelegung des Netzteilsteckers

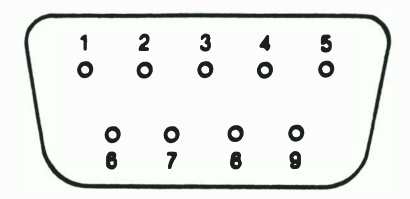
Der Control-Port 1



Pin	Bezeichnung	Bedeutung	Richtung	Erläuterung	Beschreibung in
1	JOYB0	Joystick Port B, Bit 0	Progrm.	PA0 von CIA # 1, parallel zu COL 0 der Tastatur	8.1
2	JOYB1	Joystick Port B, Bit 1	Progrm.	PA1 von CIA # 1, parallel zu COL 1 der Tastatur	8.1
3	JOYB2	Joystick Port B, Bit 2	Progrm.	PA2 von CIA # 1, parallel zu COL 2 der Tastatur	8.1
4	JOYB3	Joystick Port B, Bit 3	Progrm.	PA3 von CIA # 1, parallel zu COL 3 der Tastatur	8.1
5	POT BY	Poti Port B, Y-Richtung	Eingang	Liegt über einen Analogschalter (4066) und einen nach Masse geschalteten Kondensator (1000pF) an POTY des SID	8.2
6	BUTTON B	Feuerknopf Port B	Progrm.	PA4 von CIA # 1, parallel zu COL 4 der Tastatur	8.1
7	+5V DC	Betriebsspannung	–	Versorgungsspannung (+5 Volt) für Erweiterungen	7.5
8	GND	Ground (= Masse)	–	Systemmasse (0 Volt)	1.3.2
9	POT BX	Poti Port B, X-Richtung	Eingang	Liegt über einen Analogschalter (4066) und einen nach Masse geschalteten Kondensator (1000pF) an POTX des SID	8.2

Pinbelegung des Control-Port 1

Der Control-Port 2



Pin	Bezeichnung	Bedeutung	Richtung	Erläuterung	Beschreibung in
1	JOYB0	Joystick Port B, Bit 0	Progrm.	PA0 von CIA # 1, parallel zu COL 0 der Tastatur	8.1
2	JOYB1	Joystick Port B, Bit 1	Progrm.	PA1 von CIA # 1, parallel zu COL 1 der Tastatur	8.1
3	JOYB2	Joystick Port B, Bit 2	Progrm.	PA2 von CIA # 1, parallel zu COL 2 der Tastatur	8.1
4	JOYB3	Joystick Port B, Bit 3	Progrm.	PA3 von CIA # 1, parallel zu COL 3 der Tastatur	8.1
5	POT BY	Poti Port B, Y-Richtung	Eingang	Liegt über einen Analogschalter (4066) und einen nach Masse geschalteten Kondensator (1000pF) an POTY des SID	8.2
6	BUTTON B	Feuerknopf Port B	Progrm.	PA4 von CIA # 1, parallel zu COL 4 der Tastatur	8.1
7	+5V DC	Betriebsspannung	–	Versorgungsspannung (+5 Volt) für Erweiterungen	7.5
8	GND	Ground (= Masse)	–	Systemmasse (0 Volt)	1.3.2
9	POT BX	Poti Port B, X-Richtung	Eingang	Liegt über einen Analogschalter (4066) und einen nach Masse geschalteten Kondensator (1000pF) an POTX des SID	8.2

Pinbelegung des Control-Port 2

Anhang D
Hersteller-, Anbieter- und Literaturverzeichnis

Wenn Sie zu speziellen Bauelementen genauere Fragen haben, wenden Sie sich bitte an den jeweiligen Hersteller. In der Regel wird man gerne bereit sein, Ihnen das entsprechende Datenblatt zukommen zu lassen.
Viele Standardbauelemente werden gleich von mehreren Herstellern angeboten. Falls Sie den Produzenten nicht wissen, gibt Ihnen vielleicht die Beschriftung Auskunft. Die einzelnen Firmen kennzeichnen ihre Produkte durch verschiedene Buchstaben, die der Seriennummer vorangestellt werden. Diese Firmenbezeichnungen sind für TTL-Bausteine wie folgt:

A 0	Siliconix, Inc., USA
A 5	Insetec, USA
Am	Advanced Micro Devices (AMD), Inc., USA
CD	RCA Corp., Solid State Div., USA
CH	Cermetek, Inc., USA
CM	Solitron Devices, Inc., USA
D	Intel Corp., USA
DF, DG	Siliconix, Inc., USA
DM	National Semiconductor, Microcircuits Div., USA
DV	Hybrid Systems Corp., USA
F	Fairchild Camera and Instrument Corp., USA
GZF	R.T.C. La Radiotechnique Compelec., Frankreich
GZN	Valvo GmbH, BRD
HCC, HCF	SGS/Ates, Componenti Elettronici (S.p.A.), Italien

HD1-	Monolithic Memories, Inc., USA
HD1, HD9	Harris Semiconductor, USA
HD	Hitachi, Ltd., Japan
HEF	Valvo GmbH, BRD
HI9	Harris Semiconductor, USA
HM	Hitachi, Ltd., Japan
HM1, HM 9	Harris Semiconductor, USA
ICL, ICM, IH, IM	Intersil, Inc., USA
K	Elektronorgtechnika, UdSSR
KM	Toko America, Inc., USA
LC	Tokyo Sanyo Electric Co., Ltd., Japan
M 00	SGS/Ates, Componenti Elettronici (S.p.A.), Italien
M 5	Mitsubishi Electric Corp., Japan
MB 8	Fujitsu Ltd., Japan
MC, MC 14, MCC, MCM	Motorola Semiconductor Products, Inc., USA
MD	Mitel Semiconductor Inc., Canada
MHB	Tesla Piestany, CSSR
MK	Mostek Corp., USA
MM	National Semiconductor, Microcircuits Div., USA
MN	Mathushita Electronics Corp., Japan
MT	Mitel Semiconductor Inc., Kanada
MUX	Hybrid Systems Corp., USA
MWC, MWS	RCA Corp., Solid State Div., USA
N	Valvo GmbH, BRD
NMC	National Semiconductor, Microcircuits Div., USA
S	American Micro Devices, Inc., USA
SCP, SCL, SCM	Solid State Scientific, Inc., USA
SFF 2	Thomson-CSF, Frankreich
SIC	Siltec International, Ltd., Kanada
SN	Monolithic Memories, Inc., USA
SN	Motorola Semiconductor Products, Inc., USA
SN	Texas Instruments, Inc., USA
SYC, SYP	Syntertek, Inc., USA
T	SGS/Ates, Componenti Elettronici (S.p.A.), Italien
TA, TF, TP	Texas Instruments, Inc., USA
TC	Toshiba Corp., Japan
U 4...D, V 4...D	VEB Kombinat Mikroelektronik, DDR
UCN	Sprague Electric Co., USA
μPD	Nippon Electric Co., Japan

Hier nun die Anschriften von westdeutschen Vertretungen bzw. Distributoren der wichtigsten Hersteller:

Advanced Micro Devices, Rosenheimer Straße 143a, 8000 München 80
Analog Devices, Mozartstraße 17, 8000 München 2
Beckmann, Frankfurter Ring 115, 8000 München 40
Fairchild, Daimlerstraße 15, 8046 Garching-Hochbrück
Ferranti, Widenmazerstraße 2, 8000 München 2
Fujitsu, Lyoner Straße 44, 6000 Frankfurt/M. 71
General Instrument GmbH, Electronic 2000, Stahlgruberring 12, 8000 München 82
Hitachi Electronic Components, Hans-Pinsel-Straße 3, 8013 Haar
ITT-Intermetall, Postfach 8040, 7800 Freiburg
Intel, Seidlstraße 27, 8000 München 2
Intermetall, Hans-Bunte-Straße 19, 7500 Freiburg
Intersil, Bavariaring 38, 8000 München 15
Maxim / Spezial-Elektronik KG, Postfach 82609, 8000 München 82
Monolithic Memories GmbH, Mauerkircherstraße 4, 8000 München 80
Micro Power Systems / TA, Leonberger Straße 46, 7140 Ludwigsburg
Mitsubishi, Brandenburger Straße 40, 4030 Ratingen
MOS Technology / Commodore, Lyoner Straße 38, Frankfurt/M. 71
Mostek, Freischützstraße 92, 8000 München 81
Motorola GmbH, Arabellastraße 17, 8000 München 81
NEC Electronics GmbH, Oberrather Straße 4, 4000 Düsseldorf 30
National Semiconductor, Industriestraße 10, 8080 Fürstenfeldbruck
RCA, Justus-von-Liebig-Ring 10, 2085 Quickborn
Rockwell, Fraunhoferstraße 11a, 8033 Martinsried
SGS-Ates GmbH, Haidling 17, 8018 Grafing
Siemens AG, Balanstraße 73, 8000 München 80
Texas Instruments GmbH, Haggertystraße 1, 8050 Freising
Toshiba GmbH, Hammer Landstraße 115, 4040 Neuss 1
Valvo Bauelemente, Burchardstraße 19, 2000 Hamburg 1
Western Digital / Intraco Systeme, Am Brunner 19, 8011 Heimstetten
Zilog, Eschenstraße 8, 8028 Taufkirchen

Die Bauelemente können Sie über den Fachhandel beziehen, am besten bei einem Elektronikgeschäft in Ihrer Nähe. Bei größeren Aufträgen lohnt sich auch eine Bestellung beim Versandhandel. Hier einige der größten Anbieter für Hobbybedarf:

Bausätze: Schmotz Elektronik, Postfach 1412, 8202 Bad Aibling
Conrad Electronik GmbH, Klaus-Conrad-Straße 1, 8452 Hirschau
Völkner Electronic, Marienberger Straße 10, 3300 Braunschweig
G. Simons electronic, Meisenweg 4, 5012 Bedburg
HW Elektronik, Chaussee 79, 2000 Hamburg 19 Eimsb.

Wenn Sie sich weiter in das Gebiet der Elektronik vertiefen wollen, sind folgende Bücher empfehlenswert:

Beckmann, Ernst: Experimente, Elektronik, Köln 1977
Bernstein, Herbert: Elektronik Hobby, München 1982
Commodore-MOS Technology: Hardware Handbuch Microcomputer Familie MCS 6500, Frankfurt
Zastrow, Peter: Fernsehempfangstechnik, Frankfurt 1977

Empfehlenswerte Datenbücher für Hobbyanwender:

IWT Verlag: TTL-Taschenbuch Teil 1, München 1983
IWT Verlag: TTL-Taschenbuch Teil 2, München 1983
IWT Verlag: CMOS-Taschenbuch Band 1, München 1981

Anhang E
Tabellen und Umrechnungen

Dieser Anhang enthält einige wichtige Tabellen und Umrechnungsformeln.

Zunächst ist da die Umrechnungs- und ASCII-Tabelle. Alle 256 möglichen 8-Bit-Kombinationen sind in ihr jeweils mit der zugehörigen Hexadezimal- und Dezimalzahl enthalten. In der niederwertigen Hälfte ist jeweils auch noch das entsprechende ASCII-Zeichen aufgeführt. Dabei wurde bei Doppeldeutigkeiten durch den nationalen Zeichenvorrat das Original-ASCII-Zeichen links und das alternative deutsche Zeichen rechts aufgeführt.

Die Kombinationen der höherwertigen Hälfte (Bit 7 gesetzt) sind oft mit gerätespezifischen Sonderzeichen belegt.

Die Bedeutung der Steuerzeichen (ASCII 0-32 und 127) ist in folgender Tabelle gesondert erläutert:

Dez.	Abk.	Bedeutung	Funktion
0	NUL	Null	keine Operation
1	SOH	Start of Heading	Vorspannanfang
2	STX	Start of Text	Textanfang
3	ETX	End of Text	Textende
4	EOT	End of Transmission	Übertragungsende
5	ENQ	Enquiry	Stationsanruf
6	ACK	Acknowledge	Bestätigung
7	BEL	Bell	Klingel
8	BS	Backspace	Rückwärtsschritt
9	HT	Horizontal Tabulation	Horizontaltabulator
10	LF	Line Feed	Zeilenvorschub
11	VT	Vertical Tabulation	Vertikaltabulator
12	FF	Form Feed	Formularvorschub
13	CR	Carriage Return	Wagenrücklauf
14	SO	Shift Out	Umschalten
15	SI	Shift In	Zurückschalten
16	DLE	Data Link Escape	Austritt aus der Datenverbindung

Dez.	Abk.	Bedeutung	Funktion
17	DC1	Device Control 1	Gerätesteuerung 1
18	DC2	Device Control 2	Gerätesteuerung 2
19	DC3	Device Control 3	Gerätesteuerung 3
20	DC4	Device Control 4	Gerätesteuerung 4
21	NAC	Negative Acknowledge	Fehlermeldung
22	SYN	Synchronous Idle	Synchronisierung
23	ETB	End of Transmitted Block	Datenblockende
24	CAN	Cancel	Ungültig
25	EM	End of Medium	Datenträgerende
26	SUB	Substitute Character	Zeichen ersetzen
27	ESC	Escape	Rücksprung
28	FS	File Separator	File-Trennung
29	GS	Group Separator	Gruppentrennung
30	RS	Record Separator	Untergruppentrennung
31	US	Unit Separator	Einheitentrennung
32	SP	Space	Leerschritt
127	DEL	Delete, Rub Out	Löschzeichen

Mit Hilfe der Umrechnungstabelle lassen sich auch leicht vierstellige Hexadezimalzahlen in Dezimalwerte umrechnen. Man braucht nur die beiden höherwertigen Ziffern (Highbyte) und die beiden niederwertigen (Lowbyte) getrennt nachzusehen, und die zugehörigen Dezimalwerte in folgende Formel einzusetzen:

(Dezimalwert) = (Lowbyte) + 256 * (Highbyte)

Soll beispielsweise $COFA in eine Dezimalzahl umgewandelt werden, ergibt sich

(Dezimalwert) = $FA + 256 * $CO

also 250 + 256 * 192 = 49402

Natürlich geht das auch in der umgekehrten Richtung. Um eine Hexadezimalzahl aus einer Dezimalgröße zu erhalten, rechnet man zuerst High- und dann Lowbyte aus. Die entsprechenden Formeln sind:

(Highbyte) = INT ((Dezimalwert) / 256)

(Lowbyte) = (Dezimalwert) –((Highbyte) * 256)

Zu beiden Werten wird anschließend die Hexadezimalzahl nachgeschaut und zum vierstelligen Wert zusammengesetzt.

Auch hierzu ein Beispiel mit 49154:

(Highbyte) = INT(192,007813) = 192 = $CO

(Lowbyte) = 49154 - 49152 = 2 = $O2

Damit ergibt sich die gesuchte Hexadezimalzahl zu $COO2.

Anhang II – Zeichendarstellung/Code-Tabelle

Kürzel	Zeichen	Dez. (D)	Strg	Okt. (O)	Hex (H)	Binär (B)
NUL		000	^@	000	00	00000000
SOH	☺	001	^A	001	01	00000001
STX	☻	002	^B	002	02	00000010
ETX	♥	003	^C	003	03	00000011
EOT	♦	004	^D	004	04	00000100
ENQ	♣	005	^E	005	05	00000101
ACK	♠	006	^F	006	06	00000110
BEL	•	007	^G	007	07	00000111
BS	◘	008	^H	010	08	00001000
HT	○	009	^I	011	09	00001001
LF	◙	010	^J	012	0A	00001010
VT	♂	011	^K	013	0B	00001011
FF	♀	012	^L	014	0C	00001100
CR	♪	013	^M	015	0D	00001101
SO	♫	014	^N	016	0E	00001110
SI	☼	015	^O	017	0F	00001111
DLE	►	016	^P	020	10	00010000
DC1	◄	017	^Q	021	11	00010001
DC2	↕	018	^R	022	12	00010010
DC3	‼	019	^S	023	13	00010011
DC4	¶	020	^T	024	14	00010100
NAK	§	021	^U	025	15	00010101
SYN	▬	022	^V	026	16	00010110
ETB	↨	023	^W	027	17	00010111
CAN	↑	024	^X	030	18	00011000
EM	↓	025	^Y	031	19	00011001
SUB	→	026	^Z	032	1A	00011010
ESC	←	027	^[	033	1B	00011011
FS	∟	028	^\	034	1C	00011100
GS	↔	029	^]	035	1D	00011101
RS	▲	030	^^	036	1E	00011110
US	▼	031	^_	037	1F	00011111
		032	' '	040	20	00100000
	!	033	'!'	041	21	00100001
	"	034	'"'	042	22	00100010
	#	035	'#'	043	23	00100011
	$	036	'$'	044	24	00100100
	%	037	'%'	045	25	00100101
	&	038	'&'	046	26	00100110
	'	039	'''	047	27	00100111
	(	040	'('	050	28	00101000
	)	041	')'	051	29	00101001
	*	042	'*'	052	2A	00101010
	+	043	'+'	053	2B	00101011
	,	044	','	054	2C	00101100
	-	045	'-'	055	2D	00101101
	.	046	'.'	056	2E	00101110
	/	047	'/'	057	2F	00101111
	0	048	'0'	060	30	00110000
	1	049	'1'	061	31	00110001
	2	050	'2'	062	32	00110010
	3	051	'3'	063	33	00110011
	4	052	'4'	064	34	00110100
	5	053	'5'	065	35	00110101
	6	054	'6'	066	36	00110110
	7	055	'7'	067	37	00110111
	8	056	'8'	070	38	00111000
	9	057	'9'	071	39	00111001
	:	058	':'	072	3A	00111010
	;	059	';'	073	3B	00111011
	<	060	'<'	074	3C	00111100
	=	061	'='	075	3D	00111101
	>	062	'>'	076	3E	00111110
	?	063	'?'	077	3F	00111111

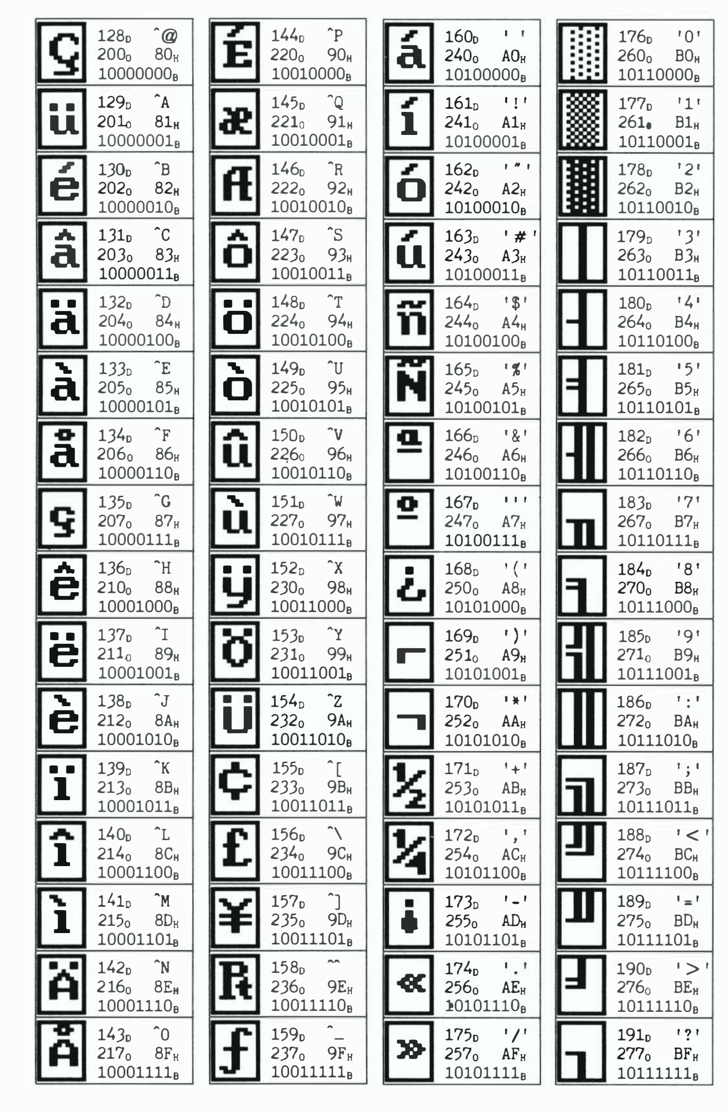
Bild II.1: *PC-Code-Tabelle mit Zeichendarstellung*

Bild II.1: *PC-Code-Tabelle mit Zeichendarstellung (Fortsetzung)*

064_D '§' 100_O 40_H 01000000_B	080_D 'P' 120_O 50_H 01010000_B	096_D ''' 140_O 60_H 01100000_B	112_D 'p' 160_O 70_H 01110000_B
065_D 'A' 101_O 41_H 01000001_B	081_D 'Q' 121_O 51_H 01010001_B	097_D 'a' 141_O 61_H 01100001_B	113_D 'q' 161_O 71_H 01110001_B
066_D 'B' 102_O 42_H 01000010_B	082_D 'R' 122_O 52_H 01010010_B	098_D 'b' 142_O 62_H 01100010_B	114_D 'r' 162_O 72_H 01110010_B
067_D 'C' 103_O 43_H 01000011_B	083_D 'S' 123_O 53_H 01010011_B	099_D 'c' 143_O 63_H 01100011_B	115_D 's' 163_O 73_H 01110011_B
068_D 'D' 104_O 44_H 01000100_B	084_D 'T' 124_O 54_H 01010100_B	100_D 'd' 144_O 64_H 01100100_B	116_D 't' 164_O 74_H 01110100_B
069_D 'E' 105_O 45_H 01000101_B	085_D 'U' 125_O 55_H 01010101_B	101_D 'e' 145_O 65_H 01100101_B	117_D 'u' 165_O 75_H 01110101_B
070_D 'F' 106_O 46_H 01000110_B	086_D 'V' 126_O 56_H 01010110_B	102_D 'f' 146_O 66_H 01100110_B	118_D 'v' 166_O 76_H 01110110_B
071_D 'G' 107_O 47_H 01000111_B	087_D 'W' 127_O 57_H 01010111_B	103_D 'g' 147_O 67_H 01100111_B	119_D 'w' 167_O 77_H 01110111_B
072_D 'H' 110_O 48_H 01001000_B	088_D 'X' 130_O 58_H 01011000_B	104_D 'h' 150_O 68_H 01101000_B	120_D 'x' 170_O 78_H 01111000_B
073_D 'I' 111_O 49_H 01001001_B	089_D 'Y' 131_O 59_H 01011001_B	105_D 'i' 151_O 69_H 01101001_B	121_D 'y' 171_O 79_H 01111001_B
074_D 'J' 112_O $4A_H$ 01001010_B	090_D 'Z' 132_O $5A_H$ 01011010_B	106_D 'j' 152_O $6A_H$ 01101010_B	122_D 'z' 172_O $7A_H$ 01111010_B
075_D 'K' 113_O $4B_H$ 01001011_B	091_D 'Ä' 133_O $5B_H$ 01011011_B	107_D 'k' 153_O $6B_H$ 01101011_B	123_D 'ä' 173_O $7B_H$ 01111011_B
076_D 'L' 114_O $4C_H$ 01001100_B	092_D 'Ö' 134_O $5C_H$ 01011100_B	108_D 'l' 154_O $6C_H$ 01101100_B	124_D 'ö' 174_O $7C_H$ 01111100_B
077_D 'M' 115_O $4D_H$ 01001101_B	093_D 'Ü' 135_O $5D_H$ 01011101_B	109_D 'm' 155_O $6D_H$ 01101101_B	125_D 'ü' 175_O $7D_H$ 01111101_B
078_D 'N' 116_O $4E_H$ 01001110_B	094_D '^' 136_O $5E_H$ 01011110_B	110_D 'n' 156_O $6E_H$ 01101110_B	126_D 'ß' 176_O $7E_H$ 01111110_B
079_D 'O' 117_O $4F_H$ 01001111_B	095_D '_' 137_O $5F_H$ 01011111_B	111_D 'o' 157_O $6F_H$ 01101111_B	127_D ' ' 177_O $7F_H$ 01111111_B

Fortsetzung
siehe nächste Seite

Bild II.1: *PC-Code-Tabelle mit Zeichendarstellung (Fortsetzung)*



128_D	^@	200_O	80_H	10000000_B
129_D	^A	201_O	81_H	10000001_B
130_D	^B	202_O	82_H	10000010_B
131_D	^C	203_O	83_H	10000011_B
132_D	^D	204_O	84_H	10000100_B
133_D	^E	205_O	85_H	10000101_B
134_D	^F	206_O	86_H	10000110_B
135_D	^G	207_O	87_H	10000111_B
136_D	^H	210_O	88_H	10001000_B
137_D	^I	211_O	89_H	10001001_B
138_D	^J	212_O	$8A_H$	10001010_B
139_D	^K	213_O	$8B_H$	10001011_B
140_D	^L	214_O	$8C_H$	10001100_B
141_D	^M	215_O	$8D_H$	10001101_B
142_D	^N	216_O	$8E_H$	10001110_B
143_D	^O	217_O	$8F_H$	10001111_B
144_D	^P	220_O	90_H	10010000_B
145_D	^Q	221_O	91_H	10010001_B
146_D	^R	222_O	92_H	10010010_B
147_D	^S	223_O	93_H	10010011_B
148_D	^T	224_O	94_H	10010100_B
149_D	^U	225_O	95_H	10010101_B
150_D	^V	226_O	96_H	10010110_B
151_D	^W	227_O	97_H	10010111_B
152_D	^X	230_O	98_H	10011000_B
153_D	^Y	231_O	99_H	10011001_B
154_D	^Z	232_O	$9A_H$	10011010_B
155_D	^[	233_O	$9B_H$	10011011_B
156_D	^\	234_O	$9C_H$	10011100_B
157_D	^]	235_O	$9D_H$	10011101_B
158_D	^^	236_O	$9E_H$	10011110_B
159_D	^_	237_O	$9F_H$	10011111_B
160_D	' '	240_O	$A0_H$	10100000_B
161_D	'!'	241_O	$A1_H$	10100001_B
162_D	'"'	242_O	$A2_H$	10100010_B
163_D	'#'	243_O	$A3_H$	10100011_B
164_D	'$'	244_O	$A4_H$	10100100_B
165_D	'%'	245_O	$A5_H$	10100101_B
166_D	'&'	246_O	$A6_H$	10100110_B
167_D	'''	247_O	$A7_H$	10100111_B
168_D	'('	250_O	$A8_H$	10101000_B
169_D	')'	251_O	$A9_H$	10101001_B
170_D	'*'	252_O	AA_H	10101010_B
171_D	'+'	253_O	AB_H	10101011_B
172_D	','	254_O	AC_H	10101100_B
173_D	'-'	255_O	AD_H	10101101_B
174_D	'.'	256_O	AE_H	10101110_B
175_D	'/'	257_O	AF_H	10101111_B
176_D	'0'	260_O	$B0_H$	10110000_B
177_D	'1'	261_O	$B1_H$	10110001_B
178_D	'2'	262_O	$B2_H$	10110010_B
179_D	'3'	263_O	$B3_H$	10110011_B
180_D	'4'	264_O	$B4_H$	10110100_B
181_D	'5'	265_O	$B5_H$	10110101_B
182_D	'6'	266_O	$B6_H$	10110110_B
183_D	'7'	267_O	$B7_H$	10110111_B
184_D	'8'	270_O	$B8_H$	10111000_B
185_D	'9'	271_O	$B9_H$	10111001_B
186_D	':'	272_O	BA_H	10111010_B
187_D	';'	273_O	BB_H	10111011_B
188_D	'<'	274_O	BC_H	10111100_B
189_D	'='	275_O	BD_H	10111101_B
190_D	'>'	276_O	BE_H	10111110_B
191_D	'?'	277_O	BF_H	10111111_B

Fortsetzung siehe nächste Seite

Bild II.1: *PC-Code-Tabelle mit Zeichendarstellung (Schluß)*

Dezimal	Zeichen	Oktal	Hex	Binär
192_D	'§'	300_O	$C0_H$	11000000_B
193_D	'A'	301_O	$C1_H$	11000001_B
194_D	'B'	302_O	$C2_H$	11000010_B
195_D	'C'	303_O	$C3_H$	11000011_B
196_D	'D'	304_O	$C4_H$	11000100_B
197_D	'E'	305_O	$C5_H$	11000101_B
198_D	'F'	306_O	$C6_H$	11000110_B
199_D	'G'	307_O	$C7_H$	11000111_B
200_D	'H'	310_O	$C8_H$	11001000_B
201_D	'I'	311_O	$C9_H$	11001001_B
202_D	'J'	312_O	CA_H	11001010_B
203_D	'K'	313_O	CB_H	11001011_B
204_D	'L'	314_O	CC_H	11001100_B
205_D	'M'	315_O	CD_H	11001101_B
206_D	'N'	316_O	CE_H	11001110_B
207_D	'O'	317_O	CF_H	11001111_B
208_D	'P'	320_O	$D0_H$	11010000_B
209_D	'Q'	321_O	$D1_H$	11010001_B
210_D	'R'	322_O	$D2_H$	11010010_B
211_D	'S'	323_O	$D3_H$	11010011_B
212_D	'T'	324_O	$D4_H$	11010100_B
213_D	'U'	325_O	$D5_H$	11010101_B
214_D	'V'	326_O	$D6_H$	11010110_B
215_D	'W'	327_O	$D7_H$	11010111_B
216_D	'X'	330_O	$D8_H$	11011000_B
217_D	'Y'	331_O	$D9_H$	11011001_B
218_D	'Z'	332_O	DA_H	11011010_B
219_D	'Ä'	333_O	DB_H	11011011_B
220_D	'Ö'	334_O	DC_H	11011100_B
221_D	'Ü'	335_O	DD_H	11011101_B
222_D	'^'	336_O	DE_H	11011110_B
223_D	'_'	337_O	DF_H	11011111_B
224_D	'`'	340_O	$E0_H$	11100000_B
225_D	'a'	341_O	$E1_H$	11100001_B
226_D	'b'	342_O	$E2_H$	11100010_B
227_D	'c'	343_O	$E3_H$	11100011_B
228_D	'd'	344_O	$E4_H$	11100100_B
229_D	'e'	345_O	$E5_H$	11100101_B
230_D	'f'	346_O	$E6_H$	11100110_B
231_D	'g'	347_O	$E7_H$	11100111_B
232_D	'h'	350_O	$E8_H$	11101000_B
233_D	'i'	351_O	$E9_H$	11101001_B
234_D	'j'	352_O	EA_H	11101010_B
235_D	'k'	353_O	EB_H	11101011_B
236_D	'l'	354_O	EC_H	11101100_B
237_D	'm'	355_O	ED_H	11101101_B
238_D	'n'	356_O	EE_H	11101110_B
239_D	'o'	357_O	EF_H	11101111_B
240_D	'p'	360_O	$F0_H$	11110000_B
241_D	'q'	361_O	$F1_H$	11110001_B
242_D	'r'	362_O	$F2_H$	11110010_B
243_D	's'	363_O	$F3_H$	11110011_B
244_D	't'	364_O	$F4_H$	11110100_B
245_D	'u'	365_O	$F5_H$	11110101_B
246_D	'v'	366_O	$F6_H$	11110110_B
247_D	'w'	367_O	$F7_H$	11110111_B
248_D	'x'	370_O	$F8_H$	11111000_B
249_D	'y'	371_O	$F9_H$	11111001_B
250_D	'z'	372_O	FA_H	11111010_B
251_D	'ä'	373_O	FB_H	11111011_B
252_D	'ö'	374_O	FC_H	11111100_B
253_D	'ü'	375_O	FD_H	11111101_B
254_D	'ß'	376_O	FE_H	11111110_B
255_D	' '	377_O	FF_H	11111111_B

Tabellenende

Stichwortverzeichnis

M

N

O

P

Q

R

S

Anleitung zur Verwendung der Platinenlayouts

Bitte prüfen Sie die Originallayouts!

Wegen technisch bedingter Abweichungen beim Druck des Buches ist es nicht immer gewährleistet, daß die Layouts hundertprozentig in Ordnung sind. Sie erkennen Fehler durch Sichtkontrolle der einzelnen Bahnen nach offensichtlich unerwünschten Überbrückungen. Eine gewollte Verbindung zwischen zwei Bahnen muß klar und deutlich sichtbar sein. Zur Kontrolle können Sie den Aufbau des Layouts mit der Baubeschreibung im Text vergleichen.

Die Layouts können auf mehrere Arten verarbeitet werden.

1. Methode

Schneiden Sie das gewünschte Layout aus und behandeln es mit Klarpaus-Spray, so daß das Papier transparent wird. Jetzt können Sie das Layout mit der bedruckten Seite auf die Platine legen und das Ganze belichten. Weitere Arbeitsanweisungen finden Sie im Anhang A.
Vorsicht, bei dieser Methode kann das Layout nur einmal verwendet werden!

2. Methode

Kopieren Sie das Layout auf Folie. Entsprechendes Material gibt's im Schreibwarenhandel unter dem Stichwort »Overhead-Folie«. Diese Vorlage läßt sich dann beliebig oft verwenden.
Aber auch hier Vorsicht. Beim Kopieren können die Leiterbahnen an Schwärze verlieren und müssen nachgebessert werden.
Sie können das Layout auch auf Papier kopieren und nach Methode 1 verfahren.

Platinenlayouts:

Spiegelverkehrt abgedruckt

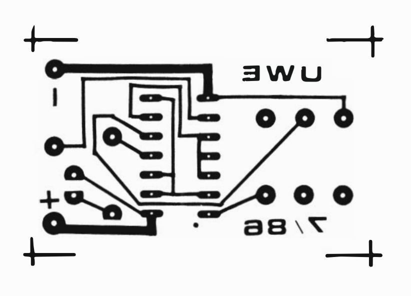


Bild 1.15

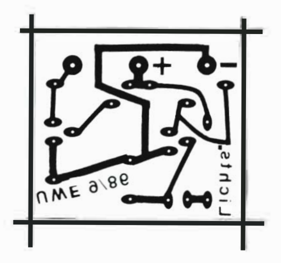


Bild 2.20

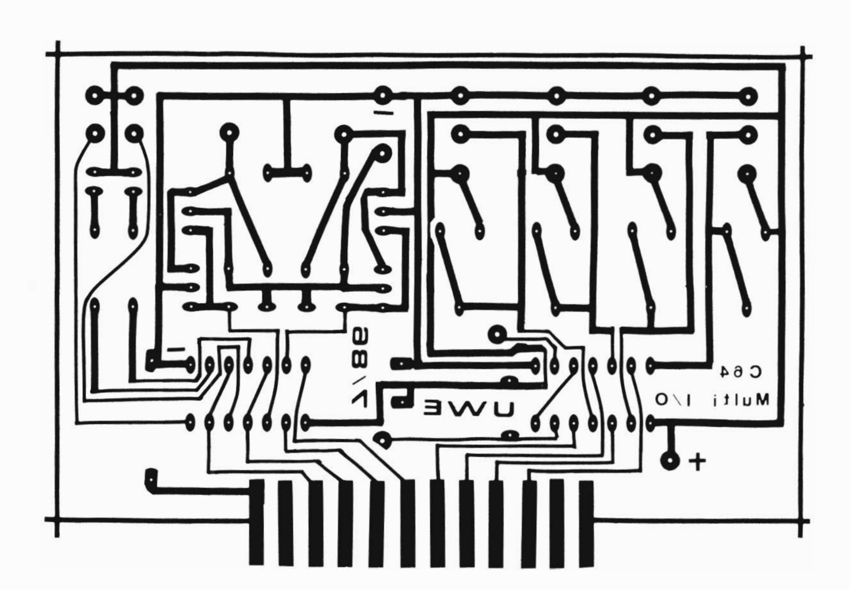


Bild 2.31

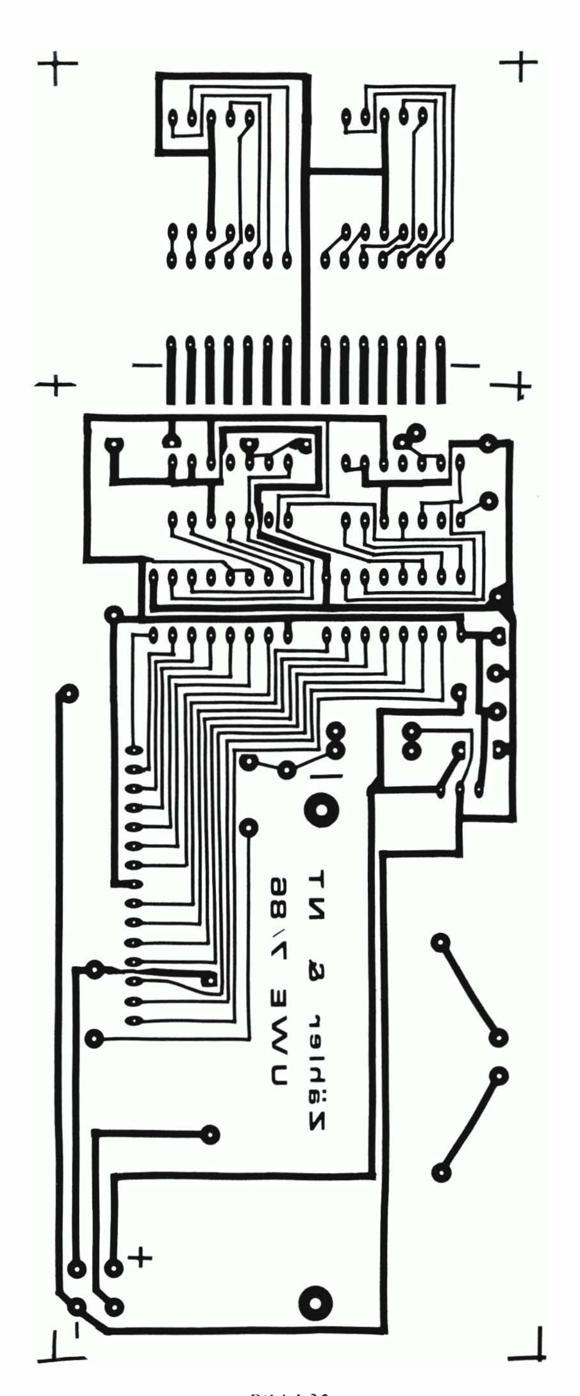


Bild 1.30

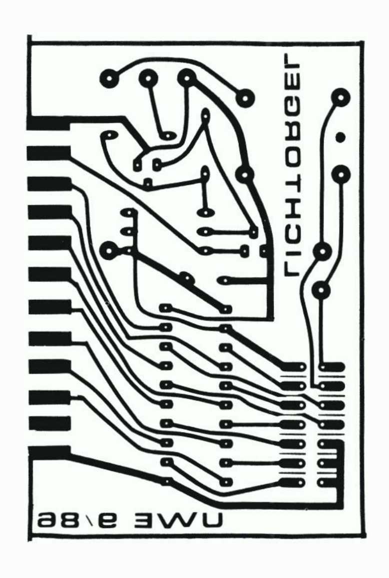


Bild 2.33

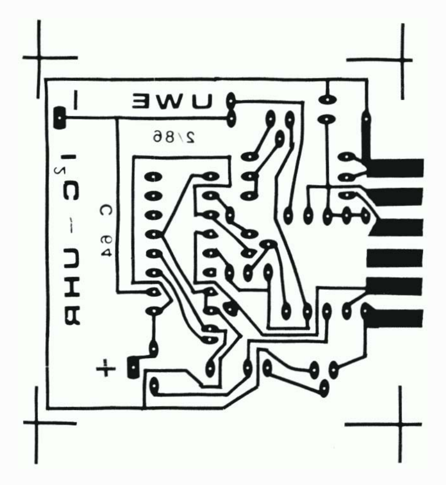


Bild 3.7

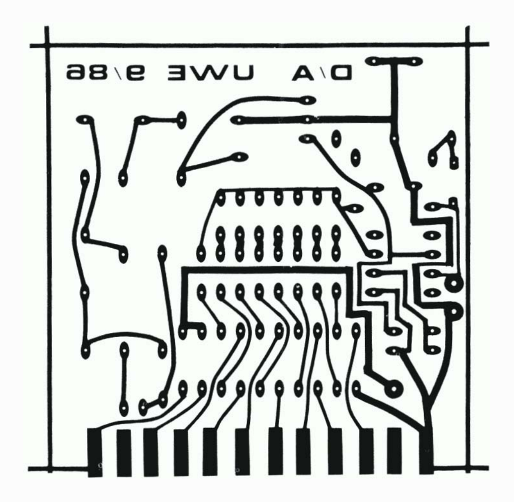


Bild 2.37

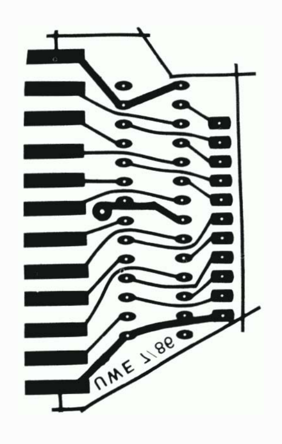
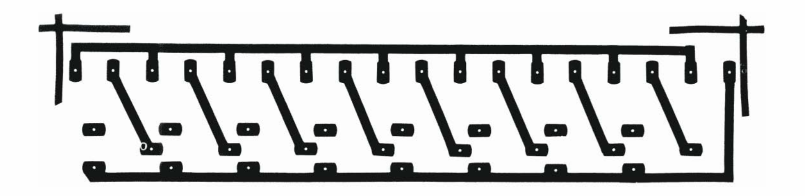


Bild 2.44



TELESCHALTER V1.1

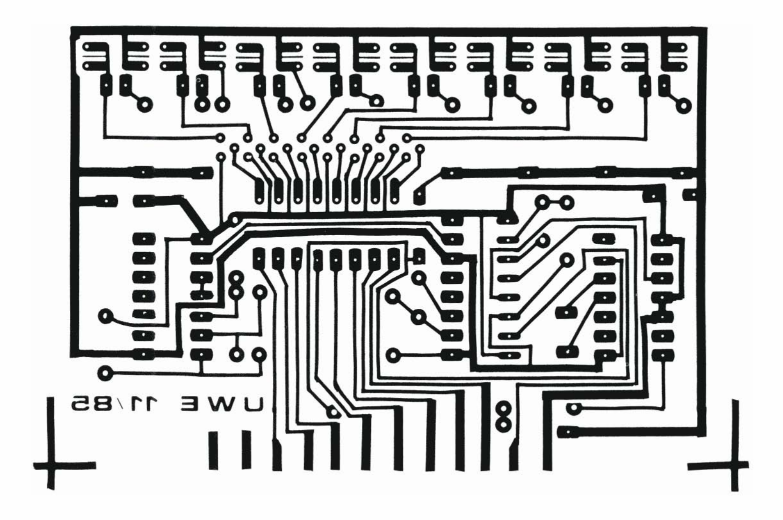


Bild 2.47

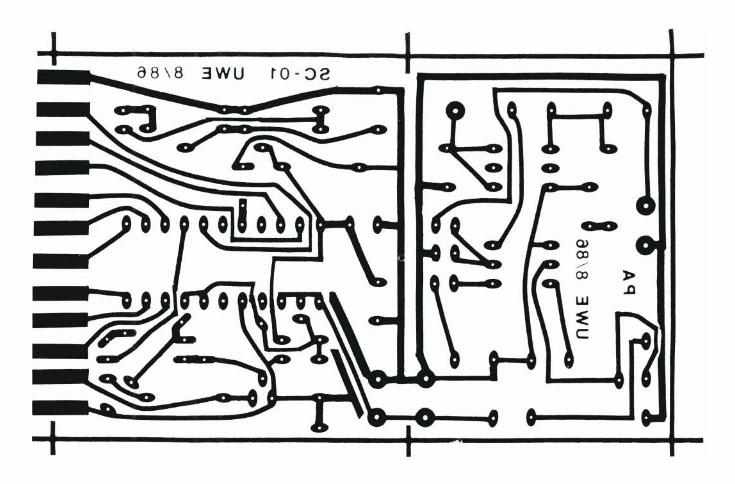


Bild 2.51

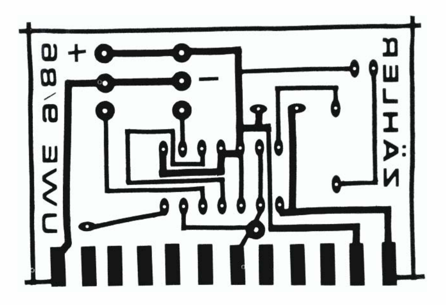


Bild 2.54

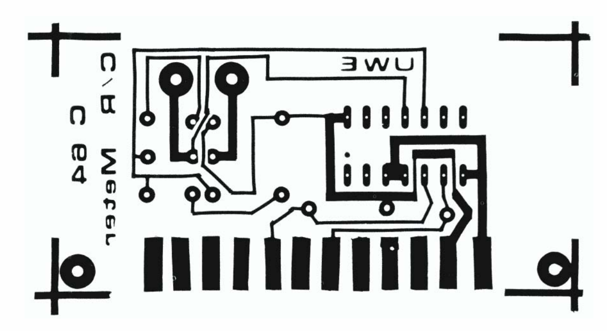


Bild 2.57

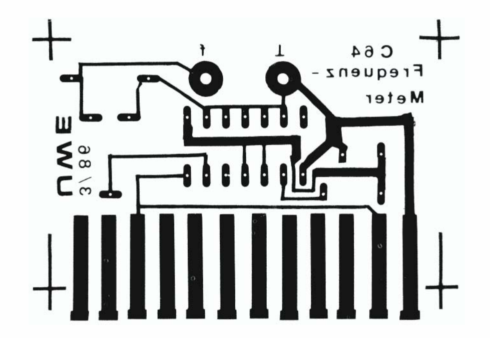


Bild 2.63

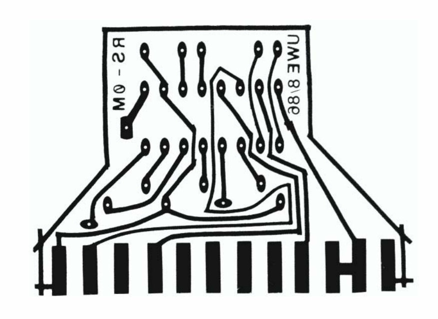


Bild 2.70

Bild 2.66

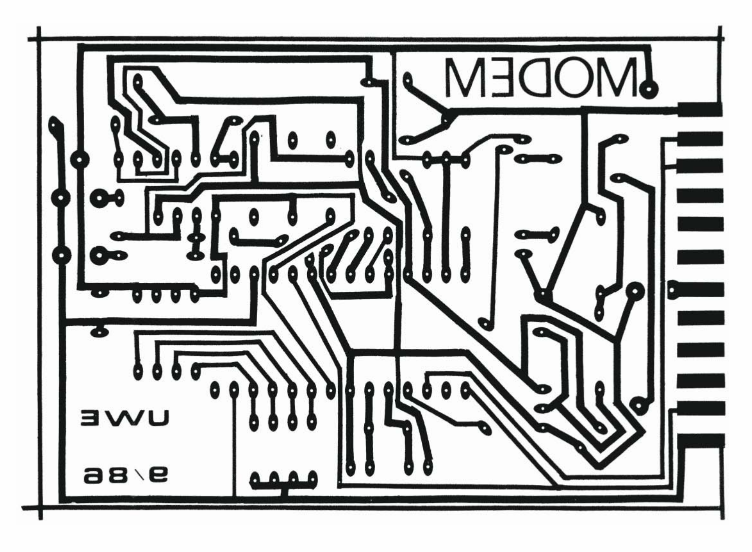


Bild 2.77

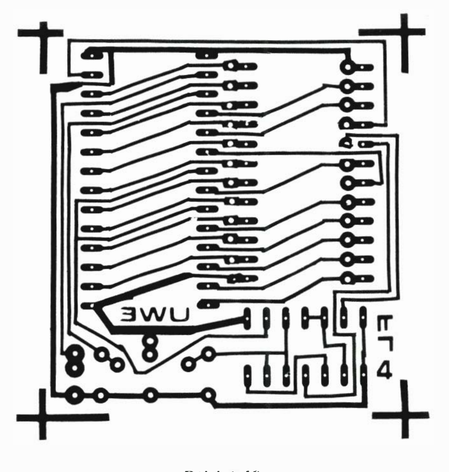


Bild 6.19

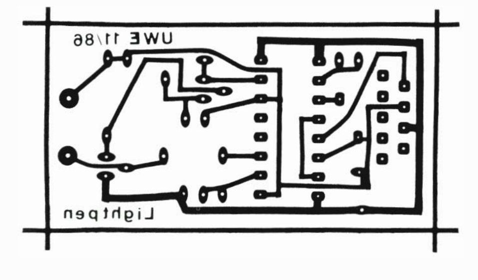


Bild 8.6

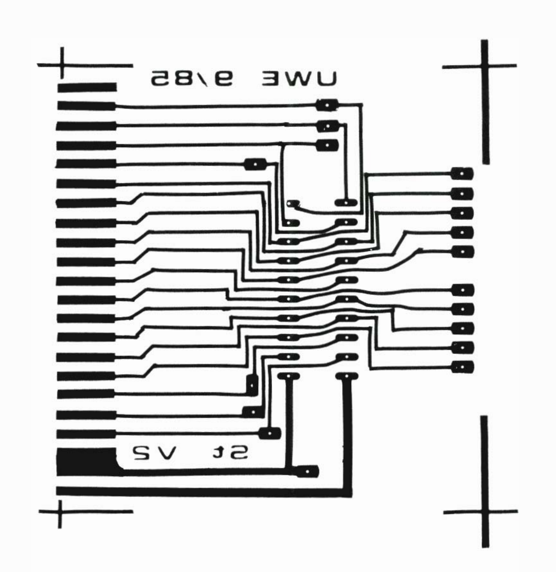


Bild 6.1 Bestückungsseite

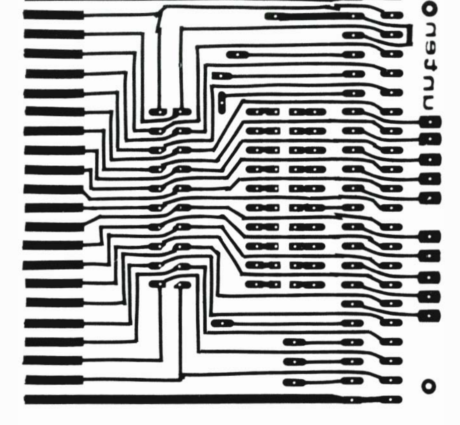


Bild 6.1 Lötseite

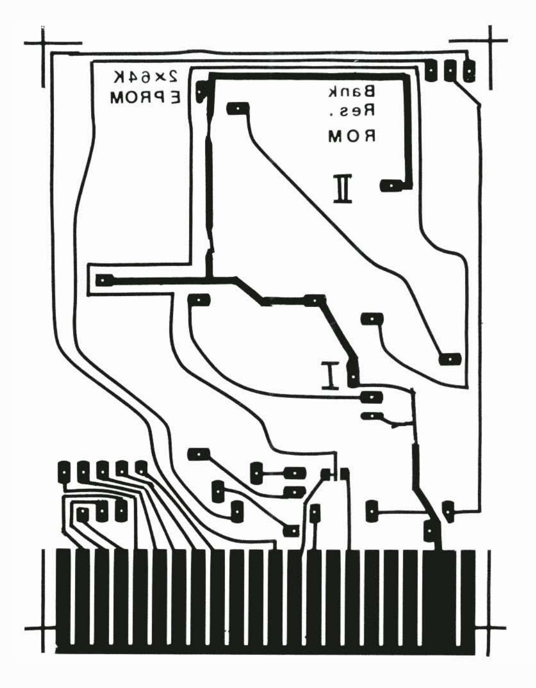


Bild 6.10 Bestückungsseite

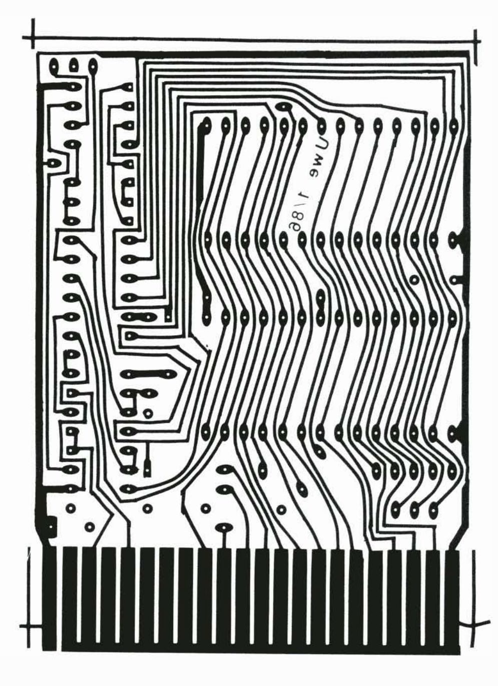


Bild 6.10 Lötseite

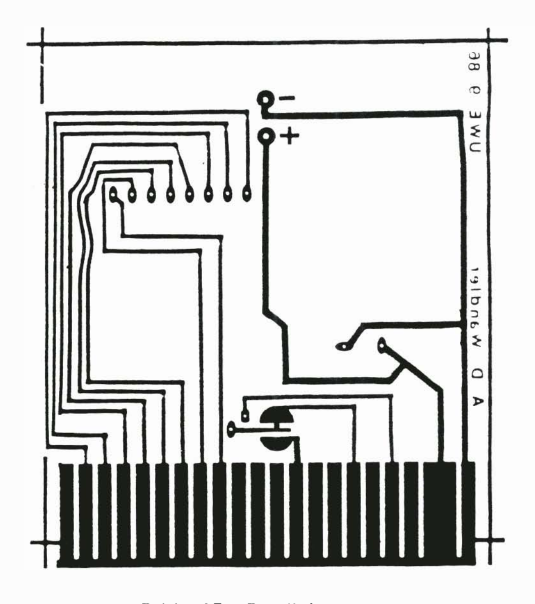


Bild 6.17 Bestückungsseite

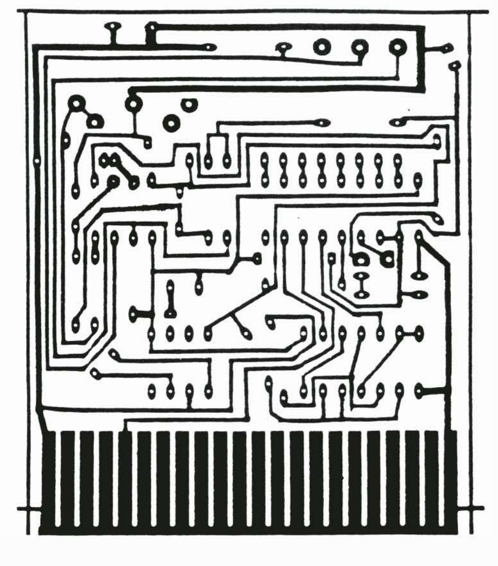
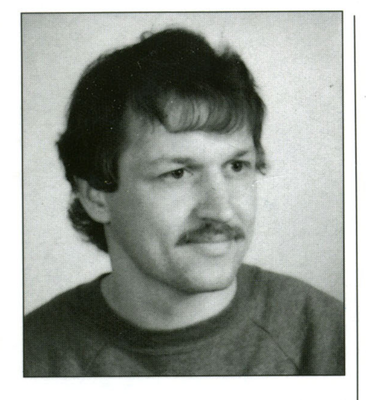


Bild 6.17 Lötseite



UWE GERLACH,
geboren am 2. März 1961, studiert Elektrotechnik an der Technischen Hochschule in Darmstadt. Sein Schwerpunkt ist die Computerhardware.
Erste Erfahrungen sammelte er bereits 1983 mit einem ZX81. Seit 1984 arbeitet der Autor mit dem C64. Sein Interesse gilt besonders den Hardware-Zusätzen, ihrer Konstruktion, ihrem Aufbau und dem praktischen Einsatz. Aus diesen Kenntnissen heraus entstanden Veröffentlichungen in diversen Fachzeitschriften und nicht zuletzt dieses Buch.

Hardware-Basteleien zum C 64/C 128

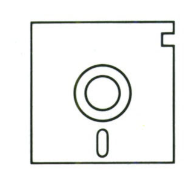
Viele C64/C128-Besitzer möchten Ihren Computer noch besser, noch weitreichender nutzen und dabei auch besser verstehen. Sie suchen nach günstigen Erweiterungen, wie z.B. D/A- bzw. A/D-Wandler, EPROM-Karte, Centronics-Treiber, Lichtschrankenmodul etc.
Dieses Buch vermittelt Ihnen die Kenntnisse, die für einen Eigenbau solcher Erweiterungen notwendig sind. Es werden die digitale Schaltungstechnik leicht verständlich erklärt, alle vorhandenen Schnittstellen des C64/C128 eingehend erläutert und ihr Einsatz anhand vieler interessanter Bauanleitungen aufgezeigt. Zu allen beschriebenen Schaltungen finden Sie genaue Selbstbauanleitungen, Stücklisten, Bezugsquellennachweise und weiterführende Literaturhinweise. Die Platinenlayouts sind im Anhang des Buches gesondert abgedruckt, um sie direkt weiterverarbeiten zu können. Auch wer sich auf Neuland begibt, findet im Anhang wichtige Tips und Tricks zum Ätzen, Löten, Bohren und Bestücken.
Neben Meß- und Steuerschaltungen sind viele nützliche Zusätze für den täglichen Betrieb des C64 enthalten.

- Entprellter Taster
- Zähler mit Netzteil
- Lichtschrankenmodul
- Multi-I/O-Platine
- Computer-Lichtorgel
- D/A-Wandler
- Centronics-Treiber
- Teleschalter
- Sprachausgabekarte
- NF-Verstärker
- Stoppuhr/Geschwindigkeitsmeßgerät
- Kapazitätsmeßgerät
- Frequenzmeter/Drehzahlmesser
- Geigerzähler
- RS232-Pegelwandler
- Weltmodem
- I2C-Echtzeituhr
- 128K-EPROM-Karte
- A/D-Wandler
- Betriebssystem-Umschaltplatine
- Paddles
- Lightpen

Im Buch enthalten ist eine Diskette im 1541-Format mit Treibersoftware.

Hardware-Voraussetzung:
C64 oder C128 (64er-Modus), Floppy 1541, 1570, 1571.





ISBN 3-89090-389-4



DM 49,– sFr 45,10 öS 382,–